Iterative Voting, Control and Sentiment Analysis

PhD Candidate: Andrea Loreggia

Supervisor: prof. Francesca Rossi

Doctoral School in Mathematical Sciences

University of Padova - Italy

UNIVERSITÀ
DEGLI STUDI
DI PADOVA

Sede Amministrativa: Università degli Studi di Padova
Dipartimento di Matematica

SCUOLA DI DOTTORATO DI RICERCA IN: SCIENZE MATEMATICHE
INDIRIZZO: COMPUTER SCIENCE
CICLO: XXVIII

ITERATIVE VOTING, CONTROL AND SENTIMENT ANALYSIS

Direttore della Scuola : Ch.mo Prof. Pier Paolo Soravia

Coordinatore d'indirizzo: Ch.mo Prof. Francesca Rossi

Supervisore :Ch.mo Prof. Francesca Rossi

Dottorando : Andrea Loreggia

Acknowledgments

I am profoundly grateful to my advisor prof. Francesca Rossi for her support and faith in my work. Thanks to her I had the chance to taste the flavour of making research, visiting and meeting great people. From her I learnt that making research is sharing ideas, being humble and listening to the others.

I would also like to acknowledge the reviewers for their valuable comments that make me improve the quality of this work.

During the last years I could work with many great researchers which inspired me and my work, for this I would like to thank prof. Toby Walsh who also hosted me at NICTA in Sydney and teach me the importance of be determined but also enjoying the work. Most of this thesis is a joint work with Brent K. Venable, Nick Mattei, Nina Narodytska, Horst Samulowitz, Yuri Malitsky, Vijay Saraswat, Cristina Cornelio, Umberto Grandi. To all these great computer scientists go my acknowledgements for what we have done together.

With all of them I worked and had fun, with all of them I share the results of this thesis. I'm really grateful for their competence in what they do, for their patience in teaching me what they know, sometimes repeating the same lesson thousands of time.

In particular, Chapter 3 describes my joint work and reports all the results proved with Nina Narodytska, prof. Francesca Rossi, Brent K. Venable and prof. Toby Walsh. Chapter 4 reports about my joint work with Umberto Grandi, prof. Francesca Rossi, Brent K. Venable and prof. Toby Walsh. Chapter 5 reports about the joint work with Umberto Grandi, prof. Francesca Rossi and Vijay Saraswat.

Finally, I'm really grateful to my family: Gabriella, Sophia and Giulia for their support

in anything I have done, for following me around the world whenever we can travel and to be always by me wherever we are.

Riassunto

Nei sistemi multi agente spesso nasce la necessità di prendere decisioni collettive basate sulle preferenze dei singoli individui. A tal fine può essere utilizzata una regola di voto che, aggregando le preferenze dei singoli agenti, trovi una soluzione che rappresenti la collettività.

In questi scenari la possibilità di agire in modo strategico può essere vista da due diversi e opposti punti di vista. Da una parte può essere desiderabile che gli agenti non abbiano alcun incentivo ad agire strategicamente, ovvero che gli agenti non abbiano incentivi a riportare in modo scorretto le proprie preferenze per influenzare il risultato dell'elezione a proprio favore, oppure che non agiscano sulla struttura del sistema elettorale stesso per cambiarne il risultato finale. D'altra parte l'azione strategica può essere utilizzata per migliorare la qualità del risultato o per incrementare il consenso del vincitore. Questi due diversi scenari sono studiati ed analizzati nella tesi. Il primo modellando e descrivendo una forma naturale di controllo chiamato "replacement control" descrivendo la complessità computazione di tale azione strategica per diverse regole di voto. Il secondo scenario è studiato nella forma dei sistemi di voto iterativi nei quali i singoli individui hanno la possibilità di cambiare le proprie preferenze al fine di influenzare il risultato dell'elezione.

Le tecniche di Computational Social Choice inoltre possono essere usate in diverse situazioni. Il lavoro di tesi riporta un primo tentativo di introdurre l'uso di sistemi elettorali nel campo dell'analisi del sentimento. In questo contesto i ricercatori estraggono le opinioni della comunità riguardanti un particolare elemento di interesse. L'opinione collettiva è estratta aggregando le opinioni espresse dai singoli individui che discutono o parlano dell’elemento di interesse attraverso testi pubblicati in blog o social network. Il lavoro di tesi studia una nuova procedura di aggregazione proponendo una nuova variante di una regola di voto ben conosciuta qual è Borda. Tale nuova procedura di aggregazione migliora le performance dell’analisi del sentimento classica.

Abstract

In multi-agent systems agents often need to take a collective decision based on the preferences of individuals. A voting rule is used to decide which decision to take, mapping the agents' preferences over the possible candidate decisions into a winning decision for the collection of agents.

In these kind of scenarios acting strategically can be seen in two opposite way. On one hand it may be desirable that agents do not have any incentive to act strategically. That is, to misreport their preferences in order to influence the result of the voting rule in their favor or acting on the structure of the election to change the outcome. On the other hand manipulation can be used to improve the quality of the outcome by enlarging the consensus of the winner. These two different scenarios are studied in this thesis. The first one by modeling and describing a natural form of control named "replacement control" and characterizing for several voting rules its computational complexity. The second scenario is studied in the form of iterative voting frameworks where individuals are allowed to change their preferences to change the outcome of the election.

Computational social choice techniques can be used in very different scenarios. This work reports a first attempt to introduce the use of voting procedures in the field of sentiment analysis. In this area computer scientists extract the opinion of the community about a specific item. This opinion is extracted aggregating the opinion expressed by each individual which leaves a text in a blog or social network about the given item. We studied and proposed a new aggregation method which can improve performances of sentiment analysis, this new technique is a new variance of a well-known voting rule called Borda.

Contents

Acknowledgments 3

Riassunto 5

Abstract 6

1 Introduction 13

1.1 Multi-mode Control 15

1.2 Iterative Voting 16

1.3 Sentiment Analysis 16

1.4 Publications 17

1.5 Thesis Structure 18

2 Background 19

2.1 Voting Systems 19

2.2 Voting Rules 20

2.3 Tie-breaking Rules 21

2.4 Voting Rules' Properties 21

2.5 Complexity 22

2.5.1 Exact cover by 3-sets (X3C) 23

2.5.2 Hitting set 23

2.6 Strategic Behaviours 23

3 Replacement Control 25

3.1 Background 27

3.2 Replacement Control 27

3.3 Relationship with Other Strategic Actions 28

3.4 Positional Scoring Rules 31

3.4.1 Plurality and veto 32

3.4.2 k-Approval 37

3.4.3 Borda 38

3.5 Approval Voting 45

3.6 Empirical Evaluation 46

4 Iterative Voting 57

4.1 Related work 60

4.2 Background 61

4.2.1 Iterative Voting 61

4.3 Restricted Iterative Voting 62

4.3.1 Restricted Iterative Voting Processes 62

4.3.2 Second-Chance and Best-Upgrade 64

4.3.3 Convergence 66

4.3.4 Quality of the Winner 69

4.4 Experimental Evaluation 71

4.4.1 Experimental Setting 71

4.4.2 Condorcet Efficiency 72

4.4.3 Borda Score 74

4.4.4 Real-world Datasets 75

4.5 Iterative Processes as a One-Stage Voting Rule 76

4.5.1 One-Stage Iterative Voting Rules 76

4.5.2 Axiomatic Properties 76

4.5.3 Computational Complexity 77

5 Sentiment analysis 79

5.1 Background 82

5.1.1 Sentiment Analysis 82

5.2 How to model individuals’ opinions 83

5.2.1 Individual data 83

5.2.2 The sentiment analysis approach 84

5.2.3 The voting theory approach 85

5.2.4 Combining Sentiment with Preference 86

5.3 Borda counts for aggregating SP-structures 87

5.3.1 Desired Axiomatic Properties 88

5.3.2 The $B^*\underline{\alpha}$ Rule 90

5.3.3 Axiomatic analysis 92

5.3.4 Algorithmic properties of $B^*\underline{\alpha}$ 94

5.4 Empirical Analysis 97

5.4.1 Analytical analysis 98

5.4.2 Sentiment Analysis and Borda Count 99

5.4.3 Incomplete Data 100

6 Conclusions 103

Bibliography 108

List of Algorithms 118

List of Figures 119

1. Introduction

During the last few decades, the trend has been for disciplines to converge on common techniques to be used in similar problems, besides focusing on specific techniques to be used in narrow domains. AI is one of the best examples: the cross-fertilisation process has led to very fascinating solutions. Consider for example genetic algorithms, which mimic evolutionary mechanisms to solve search and optimization problems [51], in this scenario solutions are evaluated using a fitness function and then they combine one each other to produce a new generation of better solutions. Or think of bird flocking or fish schooling, which are reproduced in particle swarm optimization [61] and used in coordinating autonomous driver-less cars [50].

The individualistic approach of problem solving becomes insufficient: concepts, techniques and experts need to collaborate to get a better understanding of the problems they would like to solve. The techniques that AI makes available are being used by many other disciplines. Just think of the variety of machine learning techniques used in medicine, physics or astronomy, or the constraint programming algorithms that AI researchers use to solve planning problems. AI nowadays inundates our everyday life with tools and methods that are hidden in our household electrical devices, smart-phones and much more.

Starting from the field of multi-agent systems, researchers in AI recently considered the use of models and problems from economics. Notable examples are voting systems used to aggregate the results of several search engines [30], game theoretic methods that analyse the complex interaction of autonomous agents [93], and matching procedures implemented on large-scale problems such as the coordination of kidneys transplants [1], and the assignment of students to schools [54].

In this scenario, a number of research lines federated under the name of computational social choice [90]. The need for a computational study of collective decision procedures is clear. On the one hand, from crowdsourcing to university admission ranking, many real-life applications apply existing social choice methods to large scale problems. On the other hand, collective decision-making is not a prerogative of human societies, and multi-agent systems can use these methods to coordinate their actions when facing complex situations.

A prime example is the Sydney Coordinated Adaptive Traffic System (SCATS) a real-life multi-agent system implementation used in different cities of 27 countries around the world to manage city traffic. The system uses an adaptive approach [95] which permits to adjust the management plan to the different daily traffic situations. Each intersection has a computer that manages the traffic based on an assigned plan. There are also sensors to analyse the traffic flow, this analysis allows to adjust the management of the traffic by extending or reducing the green phase. But the adjustment cannot be computed using only what a single traffic light can capture. Data from the different traffic lights of the city is sent to a central computer which produces different plausible plans. The plan is then chosen by the intersections using a voting system: each intersection votes for its preferred plan basing its preferences on what have been captured by the sensors. The plan with more preferences is chosen to manage the traffic for a specified period of time.

The thesis focuses on voting systems and aggregation methods. In particular on voting systems where the presence of strategical agents can influence the outcome of the election.

Formally, an election system $E = (C,V)$ is a tuple where C is a set of m alternatives (or candidates), V is a collection of n voters that express their preferences over the candidates, a voting rule R is used to aggregate all the preferences and choose the winner or a subset of candidates. The agent in charge of the organization of the election is called *the chair*; she can choose which candidate and/or voter can participate in the election, and which voting rule to use. A strategical agent tries to influence the outcome of the election to get a personal profit. Different kind of strategical actions can be identified based on the role played by the strategical agent in the voting system. In literature the term *manipulation* identifies the action that a voter or a coalition of voters perform on the final outcome when they report untruthful preferences. Instead, the term *control* identifies the strategical action exploited by the chair when she is interested in changing the election results in favour or against some candidate [10].

This work presents several results in three different areas connected with the computational social choice discipline. The first two line of works are related one each other because they analyze similar problems but they do that from opposite point of view. One considering strategical action as something bad to be avoided in one-shot elections and the other one considering strategical action as something beneficial that should be used in iterative election processes. Instead the third work proposes a new technique to be used in the sentiment analysis area, this technique is characterized as a new voting rule.

As already said, strategical actions are usually considered malicious because they are used to change the outcome of an election. The celebrated Gibbard-Satterthwaite theorem [49, 92] states that every voting system with more than three candidates is prone to strategic actions. It is then important to protect the system against these behaviours. In literature, the usual approach adopted against them is to design voting systems where strategical actions are very difficult to exploit and so the computational effort is so high that should not encourage anyone from using them. In this scenario we modeled and characterized a new strategical action named "replacement control" proving how computationally easy or difficult is to use it. On the other hand iterative systems allow agents to manipulate, in these systems manipulation is the engine that makes the ballot working: agents that are not satisfied by the outcome can manipulate the system to get a personal profit. In this scenarios, it is then important to propose restrictions that are computationally easy to compute and to use, and that at the same time assure the convergence to a stable state where everyone is satisfied. We also show how the outcome of these systems has a higher quality in terms for instance of larger consensus. Results on sentiment analysis represent a different approach to the computational social choice techniques. Usually sentiment analysis tries to describe the opinion of the community for a particular item, basing this decision on the individual expression. This technique works quite well when the considered item is just one, but it presents several issues when sentiment analysis is used to compared items. We studied and proposed a new aggregation method which can improve performances of sentiment analysis, this new technique is a new variance of a well-known voting rule called Borda.

1.1 Multi-mode Control

A control action may involve the deletion or addition of voters and/or candidates. While each single action has been studied in some detail (e.g. [42]), as well as its computational cost, there has not been much work on the combination of two or more control actions. But in many cases the chair could exploit many different kinds of control actions at the same time. In such a situation she could choose the action that brings the better results. The main result is from [41], which studied the combination of control actions with separated budget. That's means that each action is bounded differently. In this context, we propose a new form of multi-mode control, namely "replacement control" [70], where the same number of candidates or voters are added and deleted by the chair.

To compute the robustness of the election with respect to such actions, we study the computational complexity of the problems related to the actions.

In this line of work, we have proved some computational complexity results on well-known voting rules: plurality, veto, k-approval, Borda, approval, and Copeland, both for unweighted and for weighted voters. We have also run an extensive experimental analysis of the practical complexity of this problem to check whether such voting rules are really difficult to control in practice. To do that, we use real-world datasets from the preflib repository [72] and we run these experiment using k-approval as voting rule. Our empirical study [70] shows that plurality is more resistant to this form of control than other versions of k-approval. Moreover, we compare the control power of replacing candidates to the power of just adding or deleting them, showing that replacing candidates add significant power to the chair, with respect to to the single control action of adding or deleting candidates. For some cases where it is computational difficult, meaning that the problem is NP-complete, we also defined and run some tests to verify how difficult it is in practice, by using election from real-world dataset.

1.2 Iterative Voting

Iterative voting models an electoral process where voters are allowed to change their preferences when the outcome of the election does not satisfy them. Voters can change their preferences in order to make another more preferred candidate win the election. The process can reproduce a multi-agent system where agents cannot share their complete knowledge (in this case their preferences), either because of media limitations which do not allow to send enough information or simply because they do not trust one another. In this scenario the iterative process helps the system to reach an equilibrium where all the agents are satisfied. In this are we show some theoretical results describing under which assumptions this systems converges to a stable state where no voter has incentive to cheat, either because she is satisfied, or because she cannot affect the outcome. We also show the results of our simulations, showing that the quality of the winner after iteration is often higher than that of the winner of the initial state [53].

1.3 Sentiment Analysis

Sentiment analysis is used to classify the collective opinion about a given item [68]. This is done by extracting the individual opinions from text that individuals post on social network, such as Twitter or blogs, via natural language processing techniques. Sentiment analysis is then used to predict the opinion of the collectivity. Often it is used to predict the outcome of political elections or guessing the trend of the stock market. Sentiment analysis works quite well to predict the community opinion about a single item but it presents some issues when there are multiple items or entities to compare. We present our proposal to cope with the challenges of sentiment analysis over multiple items [52].

1.4 Publications

The work of the thesis has been presented in several international conferences. This is the list of papers related with this work classified for journal, conference or workshop where they were presented.

- Journal papers:
  - Umberto Grandi , Andrea Loreggia, Francesca Rossi, Vijay Saraswat. **A Borda count for collective sentiment analysis.** *Annals of Mathematics and Artificial Intelligence - 75 (149): 1-22, 2015.*
  - Andrea Loreggia. **Iterative voting and multi-mode control in preference aggregation.** *Intelligenza Artificiale 8 (2014) 39–51. DOI 10.3233/IA-140059. IOS Press.*
- Conference papers:
  - Andrea Loreggia, Nina Narodytska, Francesca Rossi, K. Brent Venable, and TobyWalsh. **Controlling elections by replacing candidates or votes.** *In Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS '15, pages 1737–1738, Richland, SC, 2015. International Foundation for Autonomous Agents and Multiagent Systems.*
  - Umberto Grandi, Andrea Loreggia, Francesca Rossi, Vijay Saraswat. **From Sentiment Analysis to Preference Aggregation.** *ISAIM 2014 special session on computational social choice.*
  - Umberto Grandi, Andrea Loreggia, Francesca Rossi, Kristen Brent Venable and Toby Walsh. **Restricted Manipulation in Iterative Voting: Condorcet Efficiency and Borda Score.** *In Proceedings of the 3rd International Conference on Algorithmic Decision Theory (ADT-2013), November 2013*
- Workshop papers:
  - Andrea Loreggia, Nina Narodytska, Francesca Rossi, Kristen Brent Venable and Toby Walsh. **Controlling elections by replacing candidates: theoretical and experimental results.** *In Proceedings of the 8th Multidisciplinary Workshop on Advances in Preference Handling (MPREF-2014), July 2014.*
  - Andrea Loreggia, Nina Narodytska, Francesca Rossi, Kristen Brent Venable and Toby Walsh. **Controlling elections by replacing candidates for Plurality and Veto: theoretical and experimental results.** *In Proceedings of the 1st Workshop on Exploring Beyond the Worst Case in Computational Social Choice(EXPLORE 2014), May 2014.*

- Umberto Grandi, Andrea Loreggia, Francesca Rossi. **From Sentiment Analysis to Preference Aggregation.** *R.i.C.e.R.c.A: RCRA Incontri E Confronti Workshop of the XIII AI*IA Conference.*
- Umberto Grandi, Andrea Loreggia, Francesca Rossi, Kristen Brent Venable and Toby Walsh. **Restricted Manipulation in Iterative Voting: Condorcet Efficiency and Borda Score.** *In Proceedings of the 7th Multidisciplinary Workshop on Advances in Preference Handling (MPREF-2013), August 2013.*
- Umberto Grandi, Andrea Loreggia, Francesca Rossi, Kristen Brent Venable and Toby Walsh. **Restricted Manipulation in Iterative Voting: Convergence and Condorcet Efficiency.** *In Proceeding of the 1st International Workshop on Strategic Reasoning (SR-2013), 2013.*

1.5 Thesis Structure

The thesis is divided in four main parts. The first part introduces the basic notions of voting theory used in the thesis especially the notion of voting system with a brief description about the different voting rules and the different strategical actions. The second part is dedicated to replacement control, a brief introduction introduce the motivation about this topic followed by the theoretical and empirical results. The third part reports the result about iterative voting systems, again in this case a brief introduction introduces the scenario. The fourth part describes the results related to the sentiment analysis work.

2. Background

In this section we recall the basic notions of voting theory that we shall use in this work. We briefly describe the several voting rules used as well the notion of the different strategic actions. We also recall the basic notions of computational complexity.

2.1 Voting Systems

Let C be a finite set of m candidates and V be a finite set of n individuals. We assume individuals have preferences p_i over candidates in C in the form of *strict linear orders*, i.e., transitive, anti-symmetric and complete binary relations. Individuals express their preferences in form of a ballot P_i (e.g., the top candidate, a set of approved candidates, or the full linear order) and we call the choice of a ballot for each individual a profile $P = (P_1, \dots, P_n)$. Observe that we do not allow agents to express ties among candidates, i.e., it is not possible for an agent to state that two candidates in C are equally preferred. We write $a\, P_i\, b$ to denote that agent i prefers candidate a to candidate b in profile $\mathscr{P}$, on the same way we write $a\, P_i\, C \setminus \{a\}$ to denote that agent i prefers candidate a to all the candidate in the set C. In this work, we assume that individuals submit as a ballot for the election their full linear order, and we thus use the two notions of ballot and preference interchangeably. An election E is then a pair (C,V) where C is a set of m candidates and V is a collection of n votes (linear orders over C), as already said here we assume that each voter gives a complete preference order over the set of candidates. For example given $C = \{a,b,c\}$, suppose voter v_1 prefers candidate b to a and c is her less preferred candidate, then her ballot can be represented as $v_1 : b \succ a \succ c$. As usual in the literature given an arbitrary order over a set of candidates $C = \{c_1, \ldots, c_m\}$ a preference like $v_i : C$ means that the voter i preferences respects that arbitrary order and so the preference corresponds to $v_i : a \succ b \succ c$. On the same way a preference likes $v_i : \overleftarrow{C}$ means that in the voter i preferences that arbitrary order is inverted and so we could rewrite it in the following way $v_i : c \succ b \succ a$.

2.2 Voting Rules

A (non-resolute) *voting rule* F associates with every profile $P = (P_1, \ldots, P_n)$ a non-empty subset of winning candidates $F(P) \in 2^C \setminus \{\emptyset\}$. Let us borrow from the literature some notations useful to define voting rules and later some properties [32]. In particular given two candidates c, a we set $W(c,a) = |\{i : c\ P_i\ a\}|$. There is a wide collection of voting rules that have been defined in the literature [17] and here we focus on the following ones:

Positional scoring rules (PSR): Let $(s_1, \ldots, s_m)$ be a scoring vector such that $s_1 \geqslant \ldots \geqslant s_m$ and $s_1 > s_m$. If a voter ranks candidate c at j-th position in her ballot, this gives s_j points to the candidate. The candidates with the highest score win. We focus on four particular PSR: *Plurality* with scoring vector $(1,0,\ldots,0)$, *veto* with vector $(1,\ldots,1,0)$, *k-approval* with vector $(1,1,\ldots,1,0,\ldots,0)$, where the scoring rule rewards with 1 point k candidates, and *Borda* [103] with vector $(m-1, m-2, \ldots, 0)$.

Approval: Given a subset of approved alternatives $c_i \subseteq C$ for each $i \in V$, the winners of *approval voting* are the candidates that receive the highest number of approvals.

Copeland: Any candidate c gets 1 point for each won pairwise comparison, she gets 0 point for each tie and she gets -1 point for each lost pairwise comparison. The score of c is $score(c)$=$|\{a : W(c,a) > W(a,c)\}|$-$|\{a : W(a,c) > W(c,a)\}|$.

Maximin: The score of a candidate c is the smallest number of voters preferring it in any pairwise comparison, i.e. $score(c)$=$\min_{a \in C} W(c,a)$.

Single Transferable Vote (STV): If there exists a candidate that is ranked first by the majority of the voters than this is the winner, otherwise the candidate that is ranked first by the fewest number of voters gets eliminated (ties are broken following a predetermined order of candidates). Votes initially given to the eliminated candidate are then transferred to the candidate that comes immediately after in the individual preferences. This process is iterated until one alternative is ranked first by a majority of voters.

Despite its simple definition, approval voting has been the subject of an extensive literature since its first appearance (see, e.g., [64]).

All the previous voting rules can be adapted to output a ranking of the candidates (from higher to lower score) transforming the voting rules into *social welfare functions* [90], i.e., functions which associate with every profile of preferences a ranking of the alternatives.

2.3 Tie-breaking Rules

All rules considered thus far are non-resolute, i.e., they associate a set of winning candidates with every profile of preferences.

In Chapter 3 we focus on the unique-winner model, where the agent that controls the election try to make a particular candidate p be the unique-winner of the election or she try to preclude p from being the unique-winner.

In Chapter 4 we use a tie-breaking rule to eliminate ties in the outcome. Specifically we focus on *linear tie-breaking*: the set C of candidates is ordered by $\prec_C$, and in case of ties the alternative ranked highest by $\prec_C$ is chosen as the unique outcome. Other forms of tie-breaking are possible, e.g., a random choice of a candidate from the winning set. The issue of tie-breaking has been shown to be crucial to ensure convergence of the iterative version of a voting rule [65].

2.4 Voting Rules' Properties

Every voting rule can be characterize using a set of properties. These properties can be shared among different voting rules, on the other hand different voting rules can differ one another on the properties they have. We refer to the literature for a detailed explanation of these properties (see, e.g., [90, 96]) and report a brief description of the ones used in the subsequent chapters:

Resoluteness: A voting rule F is said to be *resolute* if it always select a unique winner, i.e. $|F(P)| = 1$.

Unanimity: A voting rule F is said to be *unanimous* or *Pareto efficient* if it elects the candidate that is ranked on top of preference by all voters, when it exists, i.e. if for each agent $i \in V$, $a\, P_i\, C \setminus \{a\}$ then $F(P) = a$.

Weak Pareto: A voting rule F is said to be *weak Pareto* if it does not elect a candidate d that is dominated by another candidate c. This means that whenever for each voter $i \in V$, $c\, P_i\, d$, then $F(P) \neq d$.

Surjectivity: A voting rule F is said to be *surjective* if for each candidate $c \in C$ there exists a profile P for which $F(P) = c$.

Anonimity: A voting rule F is said to be *anonymous* if it treats individuals simmetrically, i.e. switching two individual's preferences does not change the outcome, $F(P_1,\dots,P_n)$=$F(P_{\pi(1)},\dots,P_{\pi(n)})$ where $\pi : N \to N$ is a permutation function.

Neutrality: A voting rule F is said to be *neutral* if it treats alternatives simmetrically.

Dictatorship: A voting rule F is *dictatorial* if there exists a voter $i \in V$ such that $F(P)$ is her most preferred candidate. Such a voter when it exists is called the *dictator*.

Independence of Irrelevant Alternatives: A voting rule F is *independent of irrelevant* *alternatives (IIA)* if given two candidates $c,d \in C$ and two profiles P and Q where for each voter the relative positions of c and d are the same, whenever $F(P) = c$ then $F(Q) \neq d$.

Insensitive to Bottom-ranked Candidates: A voting rule is *Insensitive to Bottom-ranked Candidates (IBC)* [63] if the winner does not change after adding or deleting a subset of candidates that all voters rank at the bottom of their preferences.

2.5 Complexity

One of the main goal of the computer science is to solve problems. For this purpose multitude of programmers and computer scientists design algorithms, which are a set of instructions useful to solve a class of problems.

The problems considered in this work are so called *decision problems* [47]. Roughly speaking this kind of problems can be thought as a question to a formal system whose answer could be either yes or no. Formally a decision problem p consists of a set I_p of *instances* and for each instance $i \in I_p$ corresponds a set $S_p(i)$ which is the set of all positive instances. An algorithm solves a decision problem if given any instance i as input of a specific problem p, the algorithm returns "no" if the set $S_p(i)$ is empty, it returns "yes" otherwise.

Sometimes we also refer to *search problems*. An algorithm is said to solve a search problem if given any instance i of the problem p it returns "no solutions" if $S_p(i)$ is empty and otherwise it returns some solutions in $S_p(i)$.

It is then important to know which is the computational effort that the algorithm takes in terms of used resources like running-time, memory and space. For this reason we say that a problem is *tractable* if there exists an algorithm which solves it taking an amount of time which is polynomial in the size of the instance. If such an algorithm does not exists then the problem is said to be *intractable*.

As usual in literature, we adopt the big-O notation ($O()$) to measure the computational complexity of an algorithm. This gives an asymptotic measure of how the algorithm responds to changes of the input size, clearly this is a worst case in the performance of the algorithm. Tractable problems have algorithms that run in $O(n^k)$, where n is the size of the input instance, which means that for large values of n the algorithm takes no more than $c \cdot n^k$ steps before ending with an answer.

We can describe the complexity of a problem based on its membership to a specific *complexity class*. There are many complexity classes that we can define but we prefer to report only the two used in this work. Specifically a decision problem is said to belong to the complexity class P if it can be solved in polynomial-time by a deterministic algorithm. Otherwise if a solution of the instance can be found by a non-deterministic algorithm (mostly because the solution is guessed among all the available ones in the solution space) and verified in polynomial time, then the problem is said to belong to *NP*. Moreover a problem is said to be *NP-hard* if its computational complexity is at least the same as the hardest problem in NP. This means that we can transform instances of a given problem into instances of another one which belongs to the same complexity class in polynomial-time with respect to the input size. This introduces the notion of *reduction* that is a function that maps each instance of a problem into an instance of another problem. If this transformation can be done in polynomial time with respect to the input size of the initial instance, then we can transfer the complexity class from the first class of problems.

In early 70s Cook lays the foundation of the modern study of the computational complexity [25]. Focusing on the *NP* complexity class, he proved that every problem in *NP* can be reduced to a particular problem name "satisfiability". Later in the 70s Karp proves that many decision problems in *NP* are equivalent in terms of computational complexity [60] and published a set of 21 equivalent problems. When a problem p is in *NP* and any other problem in *NP* can be reduced to it, then the problem is said to be *NP-complete*. Roughly speaking this means that for NP-complete problems there does not exist an efficient way to solve them, unless P=NP. This conjecture remains unproved even if it is mostly accepted to be true.

In this work we use two of the 21 NP-complete problems proposed by Karp in the 70s. For the sake of the reading we propose here the test of the problems.

2.5.1 Exact cover by 3-sets (X3C)

Given (B,S,k) an X3C instance: $B = \{b_1,\ldots,b_{3k}\}$, $S = \{S_1,\ldots,S_n\}$ such that $S_j = \{b_j^1, b_j^2, b_j^3\}, b_j^i \in B$ and a positive number $k \geqslant 1$. The question is if there exists $S' \subseteq S$ and $|S'| \leqslant k$ such that $\cup_{S_i \in S'} S_i = B$.

2.5.2 Hitting set

Given (B,S,k) an instance of the hitting set problem: $B = \{b_1,\ldots,b_n\}$, $k \leqslant n$ and $S = \{S_1,\ldots,S_m\}$ subsets of B. The question is if there exists a subset $B' \subseteq B$ with $|B'| \leqslant k$ such that B' contains at least one element from each subset of S.

2.6 Strategic Behaviours

The outcome of an election can be influenced in many different ways: for instance, voters may submit insincere preferences, whilst the chair may add or delete candidates or votes. These kinds of strategical actions are usually considered malicious and thus to prevent. One possible way to reduce the effects of these tactical behaviours is to make them very difficult to exploit and so design or use systems where the computational complexity of deciding whether the outcome can be influenced or not is very high. This may protect the election against such strategic actions. specifically we can identify in literature three different categories of strategical actions:

manipulation [11]: a voter or a coalition of voters misreport their truthful preferences to change the outcome of the election

control [10]: the chair (that is the agent in charge of the coordination of the election) acts on the structure of the election to change the outcome, this is done for instance by adding o deleting candidates or voters

bribery [42]: an external agent that cannot submit a ballot or it is not pivotal pay some voters to change their preferences with some other suggested ones

In literature, we can identify different forms of the previous strategic actions. For instance the chair can choose voting districts to favour a specific candidate or a party. This form of control is called *gerrymandering*, see for instance [56, 89].

We say that a voting rule is *immune* to a control action when the result of the election cannot be affected by it, otherwise we say that it is *susceptible* to the specified control action. If it is susceptible, we say that it is *resistant* to a control action if deciding whether it can be performed is NP-hard. On the other hand, if the decision problem is in P, we say that the system or the voting rule is *vulnerable* [42].

In the 60s Arrow proved that is impossible to design a voting system which respects a small set of desirable properties [5], i.e. there does not exist a voting system with more than two candidates which is at the same time Pareto efficient, non-dictatorship and independent of irrelevant alternatives. The original impossibility theorem was about social welfare function but this can be adapted also for voting rules. Some years later Gibbard, Satterhwaite [49, 92] and lately Duggan and Schwartz [29] showed that when a voting system has more than two candidates it can be manipulated. Usually strategic behaviour is seen as something bad that should be avoided. For this reason the first seminal works on control and manipulation [11, 10] looked at the computational complexity of the problem as a protection that the system can oppose to strategic agents. Later this line of research led to expand these results in many different ways: Studying the computational complexity of these strategic behaviour problems for different voting rules when there is only a single strategic voter [e.g. 44, 42, 96, 66, 75, 12] or by a coalition of voters [e.g. 102, 15]. The same problem was also studied parameterizing some input like for instance the number of candidates [e.g. 23, 36, 35].

3. Replacement Control

An outcome of an election can be influenced in different ways as reported in in Section 2.6. In this chapter we focus on a specific strategic behaviour called control where an external agent (usually called the chair) may decide which agents (voters) can vote and which options (candidates) can be considered. In this setting, several kinds of strategic actions can influence the result of the election. For instance, the chair may introduce new candidates or choose the voting rule. We focus here on control by the chair [10, 55].

Control may be constructive when the chair's goal is for a certain candidate to win, or destructive when it is to prevent a candidate winning. Actions that the chair can take is adding or deleting candidates or votes. Most previous works assume that only one form of control is used (see for instance, [42, 36, 35, 75]). It is natural, however, to envision the chair performing multiple control actions at the same time [41]. Here we consider a specific form of combining the basic control actions, that we call *replacement control*, where the chair replaces some candidates (or votes). This is the combination of deletion and addition of candidates (or votes) in the same quantity. A similar form of control has been considered in judgment aggregation [12].

Replacement control is a natural form of control in practice. For example, it has been widely used during the last Indian nationwide elections.[1] Replacement control is also appropriate in many parliamentary and other elections where the size of the electorate is fixed. Another setting where replacement control might occur is when the number of

[1] http://archive.indianexpress.com/news/samajwadi-party-to-replace-24-ls-candidates/1184010/, http://indiatoday.intoday.in/story/mulayam-singh-yadav-changes-a-dozen-samajwadi-party-candidates-for-2014-polls/1/326837.html

candidates (or voters) is made public. The chair can then only delete a candidate (voter) if he also adds one.

Even if replacement control seems very similar to other strategical actions, such as bribery [39] and multi-mode control [41], we show that in general it is not possible to reduce this new control action to the others, or vice-versa. This means that it is not possible to transfer the complexity results. We also relate the computational complexity of the replacement control actions to that of the single control actions of adding or deleting candidates or votes.

We study the computational complexity of replacement control for rules (plurality, veto, Borda, *k*-approval, and approval). Besides providing theoretical complexity results, where hardness informs us only about the worst case, we also performed an empirical evaluation using real-world data-sets. For some of the considered voting rules, our empirical evaluation shows that rules are easy to control despite theoretical analysis classifying the rule resistant to replacement control. These results confirm that a theoretical hardness complexity result is not enough to ensure significant protection of an election system from the control actions, as suggested also in [10, 42, 41, 40]. Thus both theoretical and empirical evidence of the hardness of control is beneficial, as argued in previous studies on manipulation [100, 99].

I have produced several papers that model and characterize replacement control. The workshop papers are more focused on the empirical study of the strategical action while the conference paper reports the theoretical details.

- Journal papers:
  - Andrea Loreggia. **Iterative voting and multi-mode control in preference aggregation.** *Intelligenza Artificiale 8 (2014) 39–51. DOI 10.3233/IA-140059. IOS Press.*
- Conference papers:
  - Andrea Loreggia, Nina Narodytska, Francesca Rossi, K. Brent Venable, and TobyWalsh. **Controlling elections by replacing candidates or votes.** *In Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS ’15, pages 1737–1738, Richland, SC, 2015. International Foundation for Autonomous Agents and Multiagent Systems.*
- Workshop papers:
  - Andrea Loreggia, Nina Narodytska, Francesca Rossi, Kristen Brent Venable and Toby Walsh. **Controlling elections by replacing candidates: theoretical and experimental results.** *In Proceedings of the 8th Multidisciplinary Workshop on Advances in Preference Handling (MPREF-2014), July 2014.*
  - Andrea Loreggia, Nina Narodytska, Francesca Rossi, Kristen Brent Venable

and Toby Walsh. **Controlling elections by replacing candidates for Plurality and Veto: theoretical and experimental results.** *In Proceedings of the 1st Workshop on Exploring Beyond the Worst Case in Computational Social Choice(EXPLORE 2014), May 2014.*

3.1 Background

In this part of the thesis most of the background is based on the notions of voting theory and computational complexity already reported in Chapter 2. For this reason in this section we describe the background that is specifically related with the control action framework.

Control actions were previously studied in a seminal paper from the 90s [10]. It builds the foundations of an area of research about the computational complexity of control action. The computational complexity is looked as a shield that the voting system can oppose to the strategic agent that tries to *constructive* change the outcome of an election in favour of a specific candidate [40]. A subsequent paper [55] introduced the *destructive* form of the control, i.e. it studies the computational complexity of the decision problem where the chair tries to ensure that a specific candidate does not win the election. Many other works expanded this line of research looking for the computational complexity of the decision problem related with the constructive or destructive form of the control using many different voting rules [e.g. 42, 91, 36, 35]. The chair cannot cast any ballot, so she can interfere with the election by acting on its structure. This can be done for instance by adding or deleting voters or by adding or deleting candidates. All these previous works accept the fact that the chair uses only one form of action to change the outcome. A subsequent work [41] modeled the situation where the chair try to use more than one control action at the same time to change the outcome of the election.

In this part of the thesis we considered the addition and deletion of candidates and/or votes. As usual in the literature, we will use the acronyms CC (for Constructive Control), DC (for Destructive Control), AC (for Adding Candidates), DC (for Deleting Candidates), AV (for Adding Votes), DV (for Deleting Votes). Although the acronym DC stands for two different notions, it will be clear from the context whether we mean Destructive Control or Deleting Candidates.

3.2 Replacement Control

Replacement control can be seen as the combination of the addition and deletion of either votes or candidates in equal amount. That is, the chair can *replace* some candidates or some votes. We use RC (for Replacing Candidates) and RV (for Replacing Votes). These will be combined with either constructive or destructive control (CC and DC). Formally, we will study the following four problems.

Name: CCRV (Constructive Control via Replacing Votes), resp., DCRV (Destructive Control via Replacing Votes)

Given: Two collections V_1, V_2 of votes, with $V_1 \cap V_2 = \emptyset$, over C, a distinguished candidate $p \in C$, and $k \in \mathbb{Z}_+$

Question (CCRV): Are there subsets $A \subseteq V_2$ and $D \subseteq V_1$ such that $|A| = |D| \leqslant k$ and p is the winner of the election $E = (C, (V_1 \setminus D) \cup A)$?

Question (DCRV): Are there subsets $A \subseteq V_2$ and $D \subseteq V_1$ such that $|A| = |D| \leqslant k$ and p is NOT the winner of the election $E = (C, (V_1 \setminus D) \cup A)$?

Name: CCRC (Constructive Control via Replacing Candidates), resp. DCRC (Destructive Control via Replacing Candidates)

Given: A collection V of votes over $C_1 \cup C_2$ (with C_1 and C_2 disjoint), a distinguished candidate $p \in C_1$, and $k \in \mathbb{Z}_+$

Question (CCRC): Are there subsets $A \subseteq C_2$ and $D \subseteq C_1$ such that $|A| = |D| \leqslant k$ and p is the winner of the election $E = ((C_1 \setminus D) \cup A, V)$?

Question (DCRC): Are there subsets $A \subseteq C_2$ and $D \subseteq C_1$ such that $|A| = |D| \leqslant k$ and $p \in (C_1 \setminus D)$ is NOT the winner of the election $E = ((C_1 \setminus D) \cup A, V)$?

We write $XY(C, A, V, p, k)$ to denote an instance of the problem, where $X \in \{CC, DC\}$, $Y \in \{AC, DC, AV, DV, RC, RV\}$, and C is a set of candidates. Moreover, if $Y \in \{AC, DC, RC\}$ then A is another set of candidates and V is the collection of votes over $C \cup A$, while if $Y \in \{AV, DV, RV\}$, then A and V are collections of votes over C. Finally, $p \in C$ is a distinguished candidate and k is the budget. Informally, A is the set of candidates or votes that the chair may add to the election, while candidates or votes to be deleted come from C or V. Notice that, when $Y \in \{DC, DV\}$, then $A = \emptyset$.

3.3 Relationship with Other Strategic Actions

Replacement control is related to multi-mode control, where the chair uses two or more control actions at the same time. However, it is not possible in general to transfer complexity results between multi-mode control and replacement control. In fact, the decision problems connected to single control actions and to replacement control could belong to different complexity classes. We just show the case for DCRC, but the proof can be easily extended to other combinations and other replacement control actions.

Theorem 3.3.1 There exists a voting rule and a class of elections on which DCAC is in P, DCDC is NP-hard and DCRC is in P.

Proof. We prove the statement by modeling a voting rule which belongs to different complexity classes with respect to the control actions. We also need to model a specific election where $m = n$, notice that every election can be massaged to be transformed in this way by either introducing dummy candidates or cancelling voters. Consider a voting rule that works like Borda when $m \geqslant n$ (where m is the number of candidates and n is the number of voters), and it works like plurality otherwise. Any election massaged to fit the class where $m = n$ is transformed in such a way that Borda and plurality results are unchanged. On this class of election, the considered voting rule is resistant to DCDC and it is vulnerable to DCAC and DCRC. In fact, plurality is resistant to DCDC [42], while Borda is vulnerable to DCAC (see Corollary 3.4.18 later in this chapter) and also to DCRC (see Theorem 3.4.17 later in this chapter). ■

Notice that replacement control is a special case of multimode control where additions equal in number the deletions. For this reason, we can inherit the polynomial results about multi-mode control but not the NP-hardness results.

Bribery is also a similar control action to replacement control. Bribery occurs when an external agent changes the preferences of some voters by bribing them with money, within a certain budget the agent has. Bribery can be seen as a special case of control where the chair replaces the vote of the bribed agent with any linear order, while in replacement control the chair can only replace a vote with another available vote. If we try to construct a polynomial many-to-one reduction from bribery to replacement control, we need to transform in polynomial time each instance of bribery to an instance of replacement control. Given an instance of bribery, we have to build all the possible linear orders using the set of candidates of the bribery instance, and this can be done in $O(m!)$, where m is the number of candidates. So we can reduce the problem of bribery to the problem of replacement control, but it seems that this cannot be done in polynomial time, unless we fix the number of candidates m as parameter. However, if we do this, for scoring rules and approval this brings the decision problems in P [39], and we cannot transfer polynomial complexity results since replacement control is a more general problem than bribery.

In general, it could be nice to find a relationship among the different strategical actions, as already done for bribery and coalitional manipulation [39], which allows to study these problems from a different point of view, but it seems that it could be done only by restricting the domain.

We now consider some axiomatic properties of voting rules and their impact on the relationship between various types of control.

Theorem 3.3.2 Every unanimous and IBC voting rule resistant to CCAC is also resistant to CCRC.

Proof. From any instance $I = CCAC(C, A, V, p, k)$ we can define an instance $I' = CCRC(C \cup D, A \cup B, V', p, k)$, where C and A are sets of candidates and these two sets are the same in the two instances I and I', D and B are two additional sets of k candidates, p is the distinguished candidate and k is the budget, V' is a collection of voters' preferences over $C \cup D \cup A \cup B$. In V and V' voters have the same preferences over $C \cup A$, moreover each voter in V' ranks unanimously on top and in the same order candidates in D and bottom-ranks and in the same order candidates in B. We claim that there exists a solution to I if and only if there exists a solution to I'. Suppose that there exists a solution $A' \subseteq A$ to I, with $|A'| = k' \leqslant k$, such that p is the unique winner in the election $(C \cup A', V)$. To make p be the unique winner of the election in I' the chair has to delete all the k candidates in D, since they are unanimously ranked. Moreover since the voting rule is IBC, candidates in B do not give any special support to any other candidate in the election, but they can be added to the election to match the number of deleted candidates, i.e. the chair can choose a subset $B' \subseteq B$ of candidates, with $|B'| = k - k'$. After the deletion of candidates in D, A' is the only subset of candidates that added to the election can make p be the unique winner. This gives a solution $(D, (A' \cup B'))$ to I'. Conversely, since the voters' preferences in I and I' are the same over $C \cup A$ it is easy to see that given a solution $(D, (A' \cup B'))$ of I' then p is also the unique winner of the election $(C \cup A', V)$. This complete the proof. ■

For the following result, we use the notion of *Insensitive to Bottom-ranked Candidates (IBC)* defined in Section 2.4. We can use the IBC property and previous results about CCDC resistance to easily define the complexity of the problem.

Theorem 3.3.3 Every voting rule that is IBC and resistant to CCDC is also resistant to CCRC.

Proof. From any instance $I = CCDC(C, \emptyset, V, p, k)$ we can derive a new instance $I' = CCRC(C, A, V', p, k)$, where C is a sets of candidates (it is the same sets in I and I'), A is a set of k fresh candidates, V' is a collection of voters' preferences over $C \cup A$, p is the distinguished candidate and k is the budget. Each voter in V' ranks at the bottom and in the same order candidates in A, while their preferences over C are the same expressed by voters in V. We claim that there exists a solution to I if and only if there exists a solution to I'. Suppose that there exists a solution $D \subseteq C$ to I, with $|D| \leqslant k$, such that p is the unique winner of the election $E = (C \setminus D, V)$. Then p is also the unique winner of the election $E' = ((C \setminus D) \cup A', V')$, with $A' \subseteq A$ and $|A'| = |D| \leqslant k$, thus giving a solution (D, A') to I'. This is proved by the fact that the addition of candidates from A does not change the winner because of IBC and preferences are the same over C, then (D, A') must be a solution of I'. Conversely, suppose that there exists a solution (D, A') to I' such that p is the unique winner of the election $E' = ((C \cup A') \setminus D, V')$. Then, p is also unique winner of the election $E = (C \setminus D, V)$, since candidates in A do not influence the winner by IBC and the voters' preferences in V and V' over C are the same. This gives us a solution D to the instance I

and complete the proof. ■

Theorem 3.3.4 Every voting rule that is IBC and resistant to DCAC or DCDC is also resistant to DCRC.

Proof. The proof is similar to that of Theorem 3.3.2 and 3.3.3. We can extend each vote with a set of dummy candidates at the bottom. Deleting/adding these candidates does not change the result. ■

3.4 Positional Scoring Rules

We now consider some results about some specific classes of voting rules. In general, positional scoring rules are vulnerable to DCRV.

Theorem 3.4.1 Positional scoring rules are vulnerable to DCRV.

Proof. Algorithm 1 is a polynomial time algorithm that checks whether it is possible to make the current winner p lose the election by replacing at most k votes. The algorithm tries to make another candidate c defeat p by adding votes that favour c the most compared to p and deleting as many votes that prefer p the most compared to c. In the pseudocode, C is the set of m candidates, V and A are collections of votes over C, p is the distinguished candidate that the function tries to make loose the election, k is the budget and s is the scoring vector which represents the positional scoring rule. Firstly, the function checks if p is the actual winner of the election or if she is already loosing the election, in this case the function stops returning two empty sets as solution. If this is not the case then the algorithm systematically checks for all candidates in $C \setminus \{p\}$, if they can defeat p. This is done by computing the support that each possible new voter $a_i \in A$ gives to p more than to each candidate $c_j \in C$. We store the information in a matrix distA where rows correspond to voters in A and column to candidates in C, $distA(a_i, c_j) < 0$ if the voter a_i prefers c_j more than p, $distA(a_i, c_j) > 0$ otherwise. Similarly, the algorithm computes the support that each voter $v_i \in V$ gives to p and store the information in another matrix distV and at the same time it computes the score difference between c_j and p, storing the information in $dist(c_j)$. Clearly $dist(c_j) < 0$ for each $c_j \in C$ since the winner of the election is p. This is done in $O(m \cdot (|A| + |V|))$. The function greedily searches for a candidate $c \in C$ that can defeat p. It starts from candidate with small distance from p and then increasingly try with all the other. For each candidate the function adds voters from A which give more support to c_j than to p and deletes voters from V that prefer p to c_j. This operation is done in time $O(k)$ to respect the budget. If p still wins after doing that, then there is no way to make p lose by replacing at most k votes. If there is a way, the algorithm returns the set of votes D to be deleted and the set A' of those to be added. Function $computeWinner(C,V,s)$ computes a winner of election V over a set of candidates C using the scoring rule s. ■

Algorithm 1 Destructive Control Replacing Votes

```
function DCRVSR($C$,$A$,$V$,$p$,$k$,$s$)
    $w \leftarrow computeWinner(C,V,s)$
    if $p \neq w$ then
        return $(\emptyset,\emptyset)$
    for all $c_j \in C$ do
        $dist(c_j) \leftarrow 0$
        for all $a_i \in A$ do
            $distA(a_i,c_j) \leftarrow s[pos(a_i,p)] - s[pos(a_i,c_j)]$
        for all $v_i \in V$ do
            $distV(v_i,c_j) \leftarrow s[pos(v_i,p)] - s[pos(v_i,c_j)]$
            $dist(c_j) \leftarrow dist(c_j) + distV(v_i,c_j)$
    $copyC \leftarrow C$
    while $copyC \neq \emptyset$ do
        $c \leftarrow \text{argmax}\, c_j \in copyCdist(c_j)$;
        $copyC \leftarrow copyC \setminus \{c\}$
        $A' = D = \emptyset$
        $copyV \leftarrow V$;
        $copyA \leftarrow A$
        for $j \leftarrow 1,k$ do
            $a \leftarrow \underset{a_i \in copyA}{\text{arg min}}\, distA(a_i,c)$;
            $v \leftarrow \underset{v_i \in copyV}{\text{arg min}}\, distV(v_i,c)$
            $A' \leftarrow A' \cup \{a\}$;
            $D \leftarrow D \cup \{v\}$
            $copyV \leftarrow copyV \setminus \{v\}$;
            $copyA \leftarrow copyA \setminus \{a\}$
            $w \leftarrow computeWinner(C,(V \setminus D) \cup A',s)$
            if $w \neq p$ then
                return $(D,A')$
    return null
```

3.4.1 Plurality and veto

Plurality and veto are two positional scoring rules. They are very similar, while plurality rewards with 1 point only the most preferred alternative in each voters' preference veto rewards all the alternatives but the last one in the preferences. In fact their scoring vectors are $(1,0,\ldots,0)$ and $((1,1,\ldots,1,0))$. They are both resistant to constructive and destructive replacing candidates, but vulnerable to constructive and destructive replacing votes.

Theorem 3.4.2 Plurality is vulnerable to CCRV.

Proof. Algorithm 2 is a polynomial-time algorithm that tries to make a candidate p win by adding as many votes as possible for p and deleting as many votes for candidates that have a higher score than p even after the additional support for p. In the pseudocode, C is the set of m candidates, V and A are collections of vote over C, p is the distinguished candidate that the function tries to make win the election, k is the budget and s is the scoring vector which represents the positional scoring rule, in this case plurality. Firstly, the function checks if p is the actual winner of the election, in this case the function stops returning two empty sets as solution. Then for each candidates $c_i \in C$ it computes the number of voters in V that most prefer c_i saving the results in a vector named score where ith position correspond to the score of c_i. The algorithm also checks if there are enough resources to make p win the election, this is done checking whether in the A set there exist enough voters that prefer p the most. After that it deletes all the voters that support candidates that have a score higher than p even after the addition of the supporters from A. At this point, if the number of voters which do not prefer p the most is higher than the budget it means that the algorithm cannot make p win the election since it already adds all the possible available support but it cannot delete enough adversaries' supporters and The function stops returning a null solution. On the other hand the algorithm can face two different scenarios: either it added more voters than it deleted or the opposite it deleted more voters than it added. In the first case the function try to match it by deleting as much voters that does not support p by randomly choosing voters which prefer candidate with high scores. In the second case the function adds voters which support candidate with small score or eventually score very different from the new score of p. If p does not win when doing that, then there is no way to make p win by replacing at most k votes. ■

Theorem 3.4.3 Plurality is vulnerable to DCRV.

Proof. Vulnerability to DCRV follows from Theorem 3.4.1 since plurality is a scoring rule. ■

Theorem 3.4.4 Plurality is resistant to CCRC.

Proof. Resistance to CCRC follows from the fact that plurality is resistant to CCDC [10] and from Theorem 3.3.3. ■

Theorem 3.4.5 Plurality is resistant to DCRC.

Proof. Resistance to DCRC follows from the fact that plurality is resistant to DCDC [55] and from Theorem 3.3.4. ■

Algorithm 2 Constructive Control Replacing Votes

```
function CCRVPL($C$,$A$,$V$,$p$,$k$,$s$)
    $w \leftarrow computeWinner(C,V,s)$
    if $p = w$ then
        return $(\emptyset,\emptyset)$
    $A' = W = \emptyset$
    for all $c_i \in C$ do
        $score(c_i) \leftarrow |\{v_j \in V : c_i = top(v_j)\}|$
    $A' \leftarrow \{a_j \in A : p = top(a_j)\}$
    $k_p = \min\{k, |A'|\}$
    $W \leftarrow \{c_j \in C : (score(c_j) - (score(p) + k_p)) \geqslant 0\}$
    $j \leftarrow 0$
    for all $w_i \in W$ do
        $listV \leftarrow \{v_j \in V : w_i = top(v_j)\}$
        while $((score(w_i) - (score(p) + k_p)) \geqslant 0)$ AND $(listV \neq \emptyset)$ do
            $v \leftarrow chooseOne(listV)$
            $D \leftarrow D \cup \{v\}$
            $listV \leftarrow listV \setminus \{v\}$
            $j \leftarrow j + 1$
            $score(w_i) \leftarrow score(w_i) - 1$
    if $j > k$ then
        return null
    while $j < k_p$ do
        $c \leftarrow \text{argmax}\, c_i \in C \setminus \{p\} score(c_i)$
        $listV \leftarrow \{v_j \in V : c = top(v_j)\}$
        if $listV \neq \emptyset$ then
            $v \leftarrow chooseOne(listV)$
            $D \leftarrow D \cup \{v\}$
            $listV \leftarrow listV \setminus \{v\}$
            $j \leftarrow j + 1$
            $score(c) \leftarrow score(c) - 1$
        else
            $C \leftarrow C \setminus \{c\}$
    while $j > k_p$ do
        $c \leftarrow \arg\min_{c_i \in C \setminus \{p\}} score(c_i)$
        $listA \leftarrow \{a_j \in A : c = top(a_j)\}$
        if $listA \neq \emptyset$ then
            $v \leftarrow chooseOne(listA)$
            $A' \leftarrow A' \cup \{v\}$
            $listA \leftarrow listA \setminus \{v\}$
            $k_p \leftarrow k_p + 1$
            $score(c) \leftarrow score(c) + 1$
        else
            $C \leftarrow C \setminus \{c\}$
    $w \leftarrow computeWinner(C, (V \cup A') \setminus D, s)$
    if $w = p$ then
        return $(D,A')$
    return null
```

Theorem 3.4.6 Veto is vulnerable to CCRV.

Proof. Algorithm 3 is a polynomial-time algorithm that checks if p can win a given election by replacing at most k votes. C is the set of candidates, V and A are multisets of voters' preferences over C, p is the distinguished candidate, k is the budget, and s is the scoring vector. The algorithm tries to make p win by deleting as many votes as possible that veto for p and adding as many votes as possible that veto for other candidates with a smaller number of veto than p even after the additional support for p. If p does not win when doing that, then there is no way to make p win by replacing at most k votes, and the algorithm returns $null$. If there is a way, the algorithm returns the set of votes D to be deleted and the set A' of those to be added. We denote $bottom(v)$ the bottom candidate in a vote v. The algorithm is very similar to the one used in the proof of Theorem 3.4.2: instead of maximizing the number of supporters for candidate p as in Algorithm 2, it tries to minimize the number of vetoes for p.

■

Theorem 3.4.7 Veto is vulnerable to DCRV.

Proof. Vulnerability to DCRV follows from Theorem 5 since veto is a scoring rule. ■

Theorem 3.4.8 Veto is resistant to CCRC.

Proof. Resistance to CCRC follows from Theorem 2 since veto is unanimous and resistant to CCAC, as shown in [66]. ■

Theorem 3.4.9 Veto is resistant to DCRC.

Proof. We prove the NP-hardness of DCRC by reduction from the hitting set problem. Given an instance $I = (B, S, k)$ of the hitting set problem, we show how to define an instance $I' = DCRC(C, A, V, p, k)$ of the DCRC problem such that I has a solution B' if and only if I' has a solution (D, A'), which means that p loses the election $((C \setminus D) \cup A', V)$. In I', C contains the following candidates: w, p, and d_j, for $j = 1, \dots, k$. We call D the set of candidates d_j. $A = \{a_1, \dots, a_n\}$ is a set of candidates the chair can use to replace candidates in C. Each candidate $a_i \in A$ correspond to an element $b_i \in B$. The collection V of votes is as follows:

$$\begin{array}{rcl}
\text{1 vote} & : & D \succ A \succ w \succ p \\
\text{2 votes for each } S_i \in S & : & p \succ D \succ A \setminus S_i \succ w \succ S_i \\
\text{2 votes for each } b_i \in B & : & p \succ D \succ w \succ A \setminus \{a_i\} \succ a_i \\
\text{2 votes for each } d_i \in D & : & p \succ w \succ A \succ D \setminus \{d_i\} \succ d_i \\
\text{2 votes for each } a_j \in A & : & p \succ w \succ D \succ A \setminus \{a_j\} \succ a_j
\end{array}$$

Algorithm 3 Constructive Control Replacing Voters

```
function CCRVVETO($C, A, V, p, k, s$)
    $w \leftarrow computeWinner(C, V, s)$
    if $p = w$ then
        return $(\emptyset, \emptyset)$
    $A' = W = \emptyset$
    for all $c_i \in C$ do
        $vetoes(c_i) \leftarrow |\{v_j \in V : c_i = bottom(v_j)\}|$
    $D \leftarrow \{v_j \in V : p = bottom(v_j)\}$
    $k_p = \min\{k, |D|\}$
    $W \leftarrow \{c_j \in C : ((vetoes(p) - k_p) - vetoes(c_j)) \geqslant 0\}$
    $j \leftarrow 0$
    for all $w_i \in W$ do
        $listA \leftarrow \{a_j \in A : w_i = bottom(a_j)\}$
        while $(((vetoes(p) - k_p) - vetoes(w_i)) \geqslant 0)$ AND $(listA \neq \emptyset)$ do
            $a \leftarrow chooseOne(listA)$
            $A' \leftarrow A' \cup \{a\}$
            $listA \leftarrow listA \setminus \{a\}$
            $j \leftarrow j + 1$
            $vetoes(w_i) \leftarrow vetoes(w_i) + 1$
    if $j > k$ then
        return null
    while $j > k_p$ do
        $c \leftarrow \arg\min_{c_i \in C} vetoes(c_i)$
        $listV \leftarrow \{v_j \in V : c = bottom(v_j)\}$
        if $listV \neq \emptyset$ then
            $v \leftarrow chooseOne(listV)$
            $D \leftarrow D \cup \{v\}$
            $j \leftarrow j + 1$
            $vetoes(c) \leftarrow vetoes(c) - 1$
            $listV \leftarrow listV \setminus \{v\}$
        else
            $C \leftarrow C \setminus \{c\}$
    while $j < k_p$ do
        $c \leftarrow \text{argmax}\, c_i \in C \setminus \{p\} vetoes(c_i)$
        $listA \leftarrow \{a_j \in A : c = bottom(a_j)\}$
        if $listA \neq \emptyset$ then
            $v \leftarrow chooseOne(listA)$
            $A' \leftarrow A' \cup \{v\}$
            $k_p \leftarrow k_p + 1$
            $vetoes(c) \leftarrow vetoes(c) + 1$
            $listA \leftarrow listA \setminus \{v\}$
        else
            $C \leftarrow C \setminus \{c\}$
    $w \leftarrow computeWinner(C, (V \cup A') \setminus D, s)$
    if $w = p$ then
        return $(D, A')$
    return null
```

We use the notation D to mean the linear order of the candidates in D, that is, $d_1, \dots, d_k$. The same is for A. Initially the number of vetoes per candidate is as follow: $veto(p) = 1$, $veto(w) = 2(m+n)$, $veto(d_i) \geqslant 2$. So p is the winner of the election and the only candidate that can defeat her is w since there is no way to reduce the number of vetoes to less than 2 for any candidate in D.

We claim that a solution to I exists if and only if there exists a solution to I'. Suppose that B' is a solution to I. Then, in I', we add the candidates in A corresponding to the elements in B' and we delete the candidates in D. This gets the following number of vetoes per candidate: $veto(p) = 1$, $veto(w) = 0$, $veto(a_j) \geqslant 4$, so p loses the election. On the other hand, suppose p loses the election by replacing at most k candidates and there exists a solution (D, A') to I'. None of the candidates added/deleted changes the vetoes for p, because of the structure of the profile, while the number of vetoes of w is decreased: for each candidate $a_i \in A$ added to the election, the number of vetoes of w decreases by 2 if $b_i \in S_i$ and no previously added candidate corresponded to another $b_j \in S_i$. Let us consider the set B' of all b_i corresponding to added candidates a_i. We have that $|B'| \leqslant k$ and contains at least one element from each subset of S. Thus it is a hitting set and therefore B' is a solution to I. ■

3.4.2 *k*-Approval

Theorem 3.4.10 For $2 < k < m-2$, k-approval is resistant to CCRV.

Proof. To prove resistance to CCRV we use a reduction from the X3C problem. We give the proof for $k = 3$, but it can be easily adapted for any k between 4 and $m-3$. Let (B, S, k) be an X3C instance: $B = \{b_1, \dots, b_{3k}\}$, $S = \{S_1, \dots, S_n\}$ such that $S_j = \{b_j^1, b_j^2, b_j^3\}, b_j^i \in B$ and a positive number $k \geqslant 1$. The question is if there exists $S' \subseteq S$ and $|S'| \leqslant k$ such that $\cup_{S_i \in S'} S_i = B$. Given an instance $I = (B, S, k)$ of the X3C problem, we build an instance $I' = CCRV(C, A, V, p, k)$, where C contains $9k+5$ candidates and V contains $(3kn - k - n)$ votes. The candidates are: p, d_1, d_2, b_j with $j = 1, \dots, 3k$, c_j^1, c_j^2, with $j = 1, \dots, 3k$, c_p^1, c_p^2.

The preferences are as follows:

$$\begin{aligned} \forall S_i \in S, 1 \text{ vote approves} &\ :\ S_i \\ \forall b_j \in B, (n - l_j) \text{votes approve} &\ :\ \{b_j, c_j^1, c_j^2\} \\ (n-k) \text{votes approve} &\ :\ \{p, d_1, d_2\} \end{aligned}$$

where l_j is the number of subsets S_i of S where b_j occurs. $A = \{a_1, \dots, a_k\}$ are the additional votes the chair may add. All these votes approve only $\{p, c_p^1, c_p^2\}$. We claim that there exists a solution to I if and only if there exists a solution to I'. Suppose that $S' = \{S_{i_1}, \dots, S_{i_k}\}$ is a solution to I. Then the chair can make p win the election by replacing all the votes in A with $D' = \{v_{i_1}, \dots, v_{i_k}\}$ votes corresponding to elements in the X3C solution. The new scores after the replacement are as follow: $score(p) = n$, $score(b_i) = n-1$, $score(c_i) = n - l_i$, $score(d_i) = n-k$, thus p is the winner and (D, A') is a solution to I'.

Let us now suppose that there exists a solution (D, A') to I', which means p wins the election $E = (C, (V \setminus D) \cup A)$. Observe that this can be achieved only by deleting k votes in a way that each candidate except p looses at least 1 point. In E, the scores are: $score(p) = n$, $score(b_i) = n - k_i$, $score(c_i) = n - l_i$, $score(d_i) = n - k$, where k_i is the number of $S_{i_j} \in S'$ in which b_i occurs. Since p is the winner of E then $score(p) > score(b_i)$ and thus $k_i > 0$. Actually, it must be $k_i = 1$ for all i. Assume $k_i > 1$ for some i and p is the winner. By a pigeon-hole argument, there is some other $k_j = 0$ with $j \neq i$. This contradicts the requirement that each $k_i > 0$. All $k_i = 1$ means that each b_i occurs exactly in one $S_{i_j} \in S'$. Thus S' is a solution to I. ■

Theorem 3.4.11 k-approval is vulnerable to DCRV.

Proof. Due to Theorem 3.4.1, k-approval is vulnerable to DCRV for all values of k. ■

The following result can be derive using Theorem B.1 from [32] and applying Theorem 3.3.2 from this work. Initially, we were not aware of this paper and we found independently a proof to the resistance of k-approval to CCRC that we report below.

Theorem 3.4.12 k-approval is resistant to CCRC.

Proof. Resistance to CCRC follows from Theorem 3.3.2 since k-approval is unanimous, IBC and resistant to CCAC, as shown in [66]. ■

Theorem 3.4.13 k-approval is resistant DCRC.

Proof. Resistance holds for DCRC due to Theorem 3.3.4 since k-approval is IBC and resistant to DCAC and DCDC, as shown in [66]. ■

3.4.3 Borda

Borda has been previously studied [91, 32] providing some results and leaving some other questions open. We prove that Borda is resistant to constructive replacement control, while it is vulnerable to its destructive versions. We also close the open problems giving new results about the single control actions of adding or deleting candidates.

Theorem 3.4.14 Borda is resistant to CCRV.

Proof. We prove the statement by giving a polynomial reduction from the X3C problem. Let (B,S,k) be an X3C instance: $B = \{b_1,\dots,b_{3k}\}$, $S = \{S_1,\dots,S_n\}$ such that $S_j = \{b_j^1, b_j^2, b_j^3\}, b_j^i \in B$ and a positive number $k \geqslant 1$. The question is if there exists $S' \subseteq S$ and $|S'| \leqslant k$ such that $\cup_{S_i \in S'} S_i = B$. Given an instance $I = (B,S,k)$ of the X3C problem, we build an instance $I' = CCRV(C,A,V,p,k')$, where $k' = 2k$. We denote $score_Y(x)$ the score of a candidate x in a subset of votes Y. The construction is based on balancing scores among the preferred candidate p, a dangerous candidate w and, also dangerous, candidates $B = \{b_i, i = 1,\dots,3k\}$ that correspond to the elements of the X3C instance. We construct an election that consists of three sets of votes: $V = V^1 \cup V^2 \cup V^3$ and a set of additional votes A. Votes in A encode subsets S_j, $j = 1,\dots,n$. The set of votes V^1, where w is always ranked first, and A, where p is always ranked first, are built in such a way that $score_V(w) - score_V(p)$ is large enough so that all votes in V^1 must be removed and exactly k votes from A must be added, otherwise p loses to w. This ensures that the entire budget must be spent. The second idea behind the reduction is to set up a score of b_i in V so that $score_V(b_i) - score_V(p)$ allows adding only one vote from A that corresponds to a set S_j that contains b_i. Therefore, the sets that correspond to k votes that are added from A must form a cover. Next we describe the reduction, in detail. The set of candidates C contains $m = 6k + (kn)^2 + 4$ candidates. More precisely, $C = \{w,p\} \cup D \cup B \cup H \cup R$, with

$$\begin{aligned}
B &= \{b_i, i = 1,\dots,3k\}\\
H &= \{h_i, i = 1,\dots,3k\}\\
R &= \{r_i, i = 1,\dots,(kn)^2\}\\
D &= \{d_1, d_2\}
\end{aligned}$$

where w is the current winner, p is the preferred candidate, b_i, $i = 1,\dots,3k$ are item-candidates, $H = \cup_{i=1,\dots,3k}\{h_i\}$, are placeholders, $R = \cup_{i=1,\dots,(kn)^2}\{r_i\}$ are separators and $\{d_1,d_2\}$ are dummy candidates. Next we describe votes. To build V we use two special pairs of votes. Each of these pairs will be applied multiple times to obtain the desired relation among scores of candidates when we form V. The first pair is $P_1(x) = \{(d_1 \succ x \succ d_2 \succ C \setminus \{d_1,d_2,x\}), (d_1 \succ \overleftarrow{C \setminus \{d_1,d_2,x\}} \succ x \succ d_2)\}$. Using this pair we ensure that the score of candidate $x \in C \setminus \{d_1,d_2\}$ increases by $m-1$, the score of candidate u, $u \in C \setminus \{d_1,d_2,x\}$ by $m-2$, the score of candidate d_2 by $m-3$, and the score of candidate d_1 by $2(m-1)$. Hence, we increase the relative score of x and $u \in C \setminus \{d_1,d_2,x\}$ by one. Using multiple copies of this pair we can set up an arbitrary difference between scores of x and u. The reason we place d_1 at the first position is to ensure that w is never ranked first when we use this construction. The second pair $P_2 = \{(d_2 \succ C \setminus \{d_1,d_2\} \succ d_1), (\overleftarrow{C \setminus \{d_1,d_2\}} \succ d_2 \succ d_1)\}$ is used to decrease the score of d_1 relative to other candidates. The votes partitioned into three logical sets: $V = V^1 \cup V^2 \cup V^3$. V^1 consists of k votes

$(w \succ B \succ H \succ R \succ D \succ p)$. Note that w is always ranked first. V^2 is the first part of the X3C encoding. To build V^2 we use a multiple copies of $P_1(x)$. Using this pair for $x \in C$, we ensure that the scores of candidates are the following:

$$\begin{aligned}
score_{V^1 \cup V^2}(p) &= c+1 \\
score_{V^1 \cup V^2}(w) &= 2k(m-1)+c \\
score_{V^1 \cup V^2}(b_i) &= k(m-1)-[(m-1)-i+(k-1)(3k-i)]+score_{V^1}(b_i)+c \\
score_{V^1 \cup V^2}(h_i) &< c \\
score_{V^1 \cup V^2}(r_i) &< c \\
score_{V^1 \cup V^2}(d_1) &> c \\
score(d_2) &< c
\end{aligned}$$

where c is the same constant in all the previous formulas.

Note that $score_{V^1 \cup V^2}(d_1) > c$ due to the choice of $P_1(x)$, where d_1 is always ranked first. The third set of votes V^3 is used to reduce the relative score between d_1 and other candidates and keeps the relative score of other candidates the same by using copies pairs of P_2: Using $score_{V^1 \cup V^2}(d_1) - c$ pairs P_2 we make sure $score_V(p) > score_V(d_1)$:

$$\begin{aligned}
score_V(p) &= c'+1 \\
score_V(w) &= 2k(m-1)+c' \\
score_V(b_i) &= k(m-1)-[(m-1)-i+(k-1)(3k-i)]+score_{V^1}(b_i)+c' \\
score_V(h_i) &< c' \\
score_V(r_i) &< c' \\
score_V(d_1) &< c' \\
score(d_2) &< c'
\end{aligned}$$

Before defining set A, we need the following suborderings. Let H be the lexicographic ordering over $\{h_1, \dots, h_{3k}\}$ and $H(t)$ be the tth element in H. We define H_{S_j} for each set $S_j = \{b_q, b_r, b_s\}$. $H_{S_j}(t) = H(t), t \neq \{q,r,s\}$ and $H_{S_j}(t) = b_t, t \in \{q,r,s\}$. In other words, H_{S_j} is H where elements $\{h_q, h_r, h_s\}$ are replaced with $\{b_q, b_r, b_s\}$. Symmetrically, let B be the lexicographic ordering over $\{b_1, \dots, b_{3k}\}$. Then, B_{S_j} is B where elements $\{b_q, b_r, b_s\}$ are replaced with $\{h_q, h_r, h_s\}$. The set A consists of n votes: $A = \cup_{j=1,\dots,n}\{(p \succ H_{S_j} \succ R \succ D \succ B_{S_j} \succ w)\}$. We now prove that I has a solution S' if and only if I' has a solution (D, A'). Assume that S' is a solution of I. We will now show that by taking $D = V^1$ and A' the subset of A which directly correspond to S', we get a solution of I'. First, V^1 must be removed. Note that $score(w) - score(p) = 2k(m-1) - 1$. Hence, w must lose $k(m-1)$ points after a replacement and p must gain $k(m-1)$ points. For w to lose $k(m-1)$ points we need to remove k votes where w is ranked first in V. The construction of V ensures that

w is only ranked first in V^1, $|V^1| = k$. Hence, we must remove all votes in V^1. Note that each b_j loses $score_{V^1}(b_i)$ points after this removal and p does not lose points. The scores after removal of V^1 are

$$\begin{aligned}
score_{V^2 \cup V^3}(p) &= c' + 1 \\
score_{V^2 \cup V^3}(w) &= k(m-1) + c' \\
score_{V^2 \cup V^3}(b_i) &= k(m-1) - [(m-1) - i + (k-1)(3k-i)] + c' \\
score_{V^2 \cup V^3}(h_i) &< c' \\
score_{V^2 \cup V^3}(r_i) &< c' \\
score_{V^2 \cup V^3}(d_1) &< c' \\
score(d_2) &< c'
\end{aligned}$$

Second, k votes from A, $A' \subset A$, must be added to the election. As there exists one-to-one correspondence between A and S, A' corresponds to S', $S' \subset S$. Let $o(b_i)$ be the number of occurrences of b_j in S'. The scores after the removal of V^1 and the addition of A' are

$$\begin{aligned}
score_{V^2 \cup V^3 \cup A'}(p) &= k(m-1) + c' + 1 \\
score_{V^2 \cup V^3 \cup A'}(w) &= k(m-1) + c' \\
score_{V^2 \cup V^3 \cup A'}(b_i) &= k(m-1) - [(1 - o(b_i))((m-1) - i) + (o(b_i) - 1)(3k - i)] + c'
\end{aligned}$$

As $o(b_i) = 1$, $i = 1, \ldots, 3k$, $score_{V^2 \cup V^3 \cup A'}(b_i) = k(m-1) + c'$. Thus p is the winner. Assume that (D, A') is a solution of I'. We now show that we can find an S' which is a solution of I. To make p the winner, D must contain V^1. Moreover, k votes from A, $A' \subset A$, must be added to the election. As there exists one-to-one correspondence between A and S, let $o(b_i)$ be the number of occurrences of b_j in S'. The scores after the removal of V^1 and the addition of A' are

$$\begin{aligned}
score_{V^2 \cup V^3 \cup A'}(p) &= k(m-1) + c' + 1 \\
score_{V^2 \cup V^3 \cup A'}(w) &= k(m-1) + c' \\
score_{V^2 \cup V^3 \cup A'}(b_i) &= k(m-1) - [(1 - o(b_i))((m-1) - i) + (o(b_i) - 1)(3k - i)] + c'
\end{aligned}$$

We have $diff(b_i, p) = score_{V^2 \cup V^3 \cup A'}(b_i) - score_{V^2 \cup V^3 \cup A'}(p) = (o(b_i) - 1)[((m-1) - i) - (3k - i)] - 1 = (o(b_i) - 1)[((m-1) - 3k] - 1$. If $o(b_j) > 1$ then $diff(b_i, p) > 0$ and p cannot be the winner. Therefore, b_j can occur at most once in S'. Since $3k$ elements from B must occur in S', $|S'| = k$. This proves that all elements in S' are distinct and form a cover, and thus a solution of I. ■

The following result can be derive using Theorem B.4 from [32] and applying Theorem 3.3.2 from this work. Initially, we were not aware of this paper and we found independently a proof to the resistance of Borda to CCRC that we report below.

Theorem 3.4.15 Borda is resistant to CCRC.

Proof. We prove the NP-hardness of the problem via reduction from X3C. Let (B,S,k) be an X3C instance: $B=\{b_1,\dots,b_{3k}\}$, $S=\{S_1,\dots,S_n\}$ such that $S_j=\{b_j^1,b_j^2,b_j^3\},b_j^i\in B$ and a positive number $k\geqslant 1$. The question is if there exists $S'\subseteq S$ and $|S'|\leqslant k$ such that $\cup_{S_i\in S'}S_i=B$. Given an instance $I=(B,S,k)$ of the X3C problem, we now build an instance $I'=CCRC(C,A,V,p,k)$ such that I' has a solution if and only if I has a solution. $C=\{p,d\}\cup B\cup D$, where B is as in I, and $D=\{d_1,\dots,d_k\}$. The set A contains n candidates: for each $S_i\in S$, there exists one candidate $a_i\in A$. V contains $12kn+18k^2-6k-2n+2$ votes. The list of their preferences is as follows. For each $S_l\in S$:

$$\begin{array}{lcl}
1 \text{ vote} & : & D\succ A\setminus\{a_l\}\succ b_l^1\succ p\succ a_l\succ d\succ B\setminus\{b_l^1\}\\
1 \text{ vote} & : & D\succ A\setminus\{a_l\}\succ b_l^2\succ p\succ a_l\succ d\succ B\setminus\{b_l^2\}\\
1 \text{ vote} & : & D\succ A\setminus\{a_l\}\succ b_l^3\succ p\succ a_l\succ d\succ B\setminus\{b_l^3\}\\
1 \text{ vote} & : & D\succ \overleftarrow{B\setminus\{b_l^1\}}\succ b_l^1\succ p\succ a_l\succ d\succ \overleftarrow{A\setminus\{a_l\}}\\
1 \text{ vote} & : & D\succ \overleftarrow{B\setminus\{b_l^2\}}\succ b_l^2\succ p\succ a_l\succ d\succ \overleftarrow{A\setminus\{a_l\}}\\
1 \text{ vote} & : & D\succ \overleftarrow{B\setminus\{b_l^3\}}\succ b_l^3\succ p\succ a_l\succ d\succ \overleftarrow{A\setminus\{a_l\}}
\end{array}$$

For each $b_i\in B$ there are $(2n+3k-2l_i-1)$ votes for each of the following preferences:

$$D\succ b_i\succ d\succ p\succ B\setminus\{b_i\}\succ A$$

$$D\succ \overleftarrow{A}\succ \overleftarrow{B\setminus\{b_i\}}\succ p\succ b_i\succ d$$

where l_i is the number of S_l in S where b_i occurs. Furthermore, there are:

$$\begin{array}{lcl}
2n+1 \text{ votes} & : & D\succ p\succ d\succ B\succ A\\
2n+1 \text{ votes} & : & D\succ \overleftarrow{A}\succ \overleftarrow{B}\succ p\succ d
\end{array}$$

We now prove that there exists a solution to I if and only if there is a solution to I'. Suppose that $S'=\{S_{i_1},\dots,S_{i_k}\}$ is a solution to I. Then the chair can make p win the election $E=((C\setminus D)\cup A',V)$. The scores in E are as follow:

$$\begin{array}{rcl}
score(p) & = & C'+3k+1\\
score(b_i) & = & C'+3k\\
score(d) & = & C'-11n\\
score(a_i) & = & C'-3
\end{array}$$

where $C'=(4k+1)((3k-1)(2n+3k)+n+1)+2n$. Thus p is the winner and (D,A') is a solution to I'.

For the reverse, suppose that there exists a solution (D,A') to I'. This means that p wins the election $E=((C\setminus D)\cup A',V)$. Observe that this can be achieved only by deleting all the candidates in D and then by adding k candidates in A, since $|D|=k$. Let $A'=\{a_{i_1},\dots,a_{i_k}\}$ be the set of added candidates. Then there exists a set $S'=\{S_{i_1},\dots,S_{i_k}\}$ such that its elements correspond to elements in A'. Scores in E are as follow:

$$\begin{aligned}
score(p) &= C'+3k+1\\
score(b_i) &= C'+3k-1+c_i\\
score(d) &= C'-11n\\
score(a_i) &= C'-3
\end{aligned}$$

where c_i is the number of $S_{i_j}\in S'$ in which b_i occurs. Since p is the winner of the new election then $score(p)>score(b_i)$ and so $C'+3k+1>C'+3k-1+c_i$, thus $c_i<2$. We now show that no c_i can be equal to zero. Assume that some $c_i=0$ and p is the winner. By the pigeon-hole principle, there must be some other $c_j\geqslant 2$ for some $j\neq i$. This contradicts the requirement that each $c_i<2$. So, each $c_i=1$, which means that each b_i occurs exactly in one $S_{i_j}\in S'$. That corresponds to an X3C and S' is a solution to I. ■

Corollary 3.4.16 Borda is resistant to CCAC.

Proof. The proof is very similar to the one of Theorem 3.4.15, Indeed, we can follow the same proof, but since there are no candidates to delete we can omit the set D. ■

We have shown that DCRC is NP-hard for plurality and veto. Surprisingly, DCRC is polynomial for Borda. This is because the difference between two consecutive scores in the Borda's scoring vector are identical, unlike in plurality or veto.

Theorem 3.4.17 Borda is vulnerable to DCRC and DCRV.

Proof. Vulnerability to DCRV follows from Theorem 3.4.1 since Borda is a scoring rule. Algorithm 4 is a polynomial algorithm that checks if p can lose a given election by replacing at most k candidates. C and A are sets of candidates, V is a multiset of voters' preferences over $C\cup A$, p is the distinguished candidate, k is the budget. The algorithm computes the Borda winner w. If p is different from w, then p already loses the election, so nothing needs to be done. Otherwise, we compute $dist(x,c_i)$, the score that c_i gives to x compared to that it gives to w. This only depends on relative positions of x, c_i and w as the scoring vector for Borda rule satisfies the property $s_i-s_{i-1}=1$ for all i. Then we consider each candidate x and check whether x can beat p by removing, one at a time, candidates that bring the smallest number of points to x and adding candidates that bring the largest number of points to x. ■

Algorithm 4 Destructive Control Replacing Candidates

```
function DCRCBORDA($(C, A, V, p, k)$)
    $w \leftarrow$Compute-Borda-winner$(C, V)$
    if $p \neq w$ then
        return $(\emptyset, \emptyset)$
    for all $x \in C \cup A$ do
        for all $c_i \in ((C \cup A) \setminus \{x\})$ do
            $dist(x, c_i) \leftarrow 0$
            for all $v_l \in V$ do
                $dist(x, c_i) \leftarrow sign(sign(pos(v_l, x) - pos(v_l, c_i)) - sign(pos(v_l, w) - pos(v_l, c_i))) + dist(x, c_i)$
    for all $x \in C \cup A$ do
        $A' = D = \emptyset; copyC \leftarrow C; copyA \leftarrow A$
        $j \leftarrow 0$
        if $x \in A$ then
            $A' \leftarrow A' \cup \{x\}; j \leftarrow 1$
            $c \leftarrow \arg\min_{c_i \in copyC} dist(x, c_i); copyC \leftarrow copyC \setminus \{c\}$
            $D \leftarrow D \cup \{c\}$
        while $j < k$ do
            $c \leftarrow \arg\min_{c_i \in copyC} dist(x, c_i); a \leftarrow \text{argmax}\, a_i \in copyAdist(x, a_i)$
            $A' \leftarrow A' \cup \{a\}; D \leftarrow D \cup \{c\}$
            $w \leftarrow$Compute-Borda-winner$((C \setminus D) \cup A', V)$
            $copyC \leftarrow copyC \setminus \{c\}; copyA \leftarrow copyA \setminus \{a\}; j \leftarrow j + 1$
            if $w \neq p$ then
                return $(D, A')$
    return $null$                                                    ▷
```

As a corollary of the above theorem, Borda is also vulnerable to adding or deleting candidates for destructive control.

Corollary 3.4.18 Borda is vulnerable to DCAC and DCDC.

Proof. The algorithm is very similar to the one in Theorem 3.4.17 except that for DCDC problem $A = \emptyset$ and, for DCAC, we do not need to find any candidate to put in the set D. ■

3.5 Approval Voting

Approval voting is resistant to CCRV and vulnerable to the other forms of replacement control.

Theorem 3.5.1 Approval is resistant to CCRV.

Proof. We prove the NP-hardness via reduction from the hitting set problem. Given an instance $I = (B,S,k)$ of the hitting set problem (defined earlier), we define an instance $I' = CCRV(C,A,V,p,k)$ of the CCRV problem such that I has a solution B' if and only if I has a solution (D,A'), which means that p wins the election $((C \setminus D) \cup A', V)$. In the instance I', $C = \{c_1,\dots,c_m\} \cup \{p\}$. Each candidate $c_i \in C \setminus \{p\}$ corresponds to an element $S_i \in S$ of the hitting set problem. The collection of votes is specified only with the list of candidates that each of them approves. The total number of votes in V is $n(m+2) - k - \sum_{S_i \in S} |S_i|$. There is a set of n votes which correspond to the elements of set B. For each such vote v_i we define $C_i = \{S_j \in S : b_i \in S_j\}$, and v_i approves all the candidates in C_i. The complete set of votes is defined as follows:

$$\begin{aligned} \forall b_i \in B,\ 1 \text{ vote approves} &\quad C_i \\ \forall S_i \in S (n - |S_i|) \text{ votes approve} &\quad c_i \\ (n-k) \text{ votes approve} &\quad p \end{aligned}$$

$A = \{a_1,\dots,a_k\}$ are the additional votes the chair may add. All these votes approve only p.

In the election (C,V), scores for the candidates are as follows: $score(p) = n-k$ and , for all $i \in \{1,\dots,m\}$, $score(c_i) = n$. We claim that p can win this election if and only if there exists a solution to the hitting set problem. Let us suppose that there exists a solution to the hitting set problem, say $\{b_{i_1},\dots,b_{i_k}\}$. Then the chair can make p win the election replacing the k votes corresponding to $\{b_{i_1},\dots,b_{i_k}\}$ with the k votes in A. The new scores are: $score(p) = n$ and $score(c_i) \leqslant n-1$. If instead p wins the election, then the chair replaces votes so that $score(p) > score(c_i)$, for each $c_i \in C$. The maximal number of points that p can gain is $|A| = k$. That means each $c_i \in C$ has to lose at least 1 point. Thus the chair needs to eliminate k votes that support all c_i's. These eliminated votes correspond to elements of B which are a solution to the hitting set problem. If it were not a solution, then some candidates are not supported by these votes, so the elimination of such votes does not change the score of some c_i. Thus, c_i still has n points and p is not the unique winner of the election which is a contradiction. ■

Theorem 3.5.2 Approval is vulnerable to CCRC, DCRC and DCRV.

Proof. To prove vulnerability to CCRC we can just compute the score of all candidates. To check whether p can be the unique winner by replacing candidates, we can check if every candidate in C that has a score higher or equal to the score of p (i.e., candidates in C which are approved by more voters or by the same number of voters than p) can be replaced with another candidate in A that has a score that is smaller than the score of p. This can be done in $O(|V| \times |C| \times |A|)$. Let $D \subseteq C$ be the subset of candidates with score equal or higher than the score of p and $A' \subseteq A$ the subset of candidates with score smaller than the score of p, the distinguished candidate p cannot be the unique winner if: $|D| > k$ (i.e. the chair has not enough budget to replace all the candidates that have more support than p) or $|D| > |A'|$ (i.e. there are not enough resources to exploit a replacement). If $|D| < k$ and $|D| \leqslant |A'|$ then p can be the unique winner. Similarly we can prove the vulnerability to DCRC by computing the score of all candidates in A and replacing a candidate in $C \setminus \{p\}$ with someone that has a score higher or equal to p. If such a candidate does not exist, then p cannot be defeated. To prove the vulnerability to DCRV, we can use a modified version of the algorithm used in the proof of Theorem 3.4.1 where, for each vote, we use a vector of 0s and 1s (where 0 stands for not approved and 1 stands for approved) to describe the set of candidates that the vote is approving, rather than using a scoring vector which is the same for all votes. ■

3.6 Empirical Evaluation

To better understand the theoretical results showing the hardness or easiness of the DCRC control problem, we performed an empirical evaluation on real-world datasets. This allows one to understand if a certain replacement control problem is really difficult in practice. Besides plurality and veto, in this experimental analysis we also consider k-approval for values of k that differ from 1 to $m-1$ as this naturally interpolates between plurality (1-approval) and veto (m-1-approval). We also run some experiments using Borda, to check whether in practice it is easy, as the theoretical results say.

We focus on the DCRC problem in k-approval. given a set C of qualified candidates, a set A of unqualified candidates and the voters' preferences. Doing this we assume that the

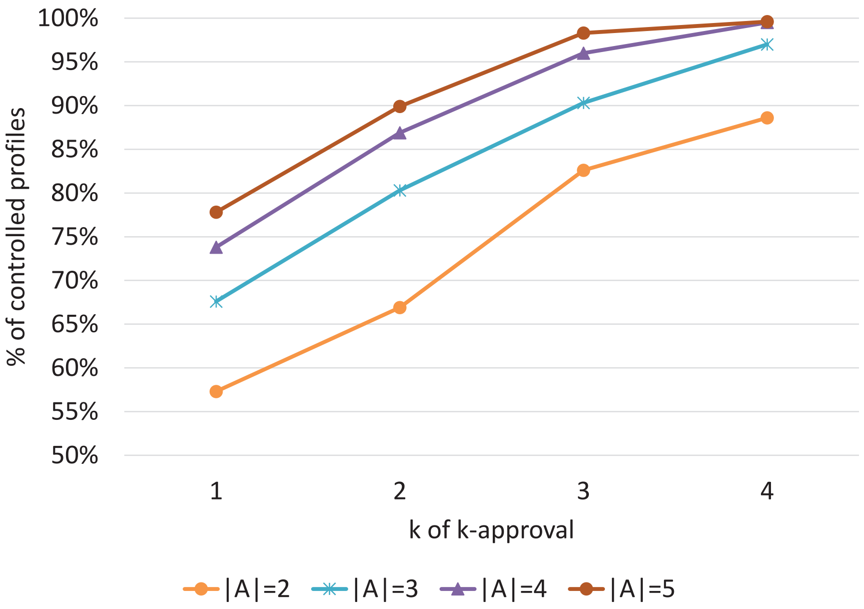

Figure 3.1: Sushi dataset: fraction of profiles (over 1000) with successful DCRC.

budget is $k = |A|$. We also compare DCRC with single control actions which just add or delete candidates (DCAC and DCDC).

We consider profiles coming from real world data sets. In particular, we use three datasets from the prelib repository (www.preflib.org) [72]:

- the AGH Course Selection ED00009, which contains preferences of some university students over a set of courses [94];
- the T-Shirt ED00012 dataset, which contains preferences of some NICTA employees over some t-shirt templates;
- the sushi dataset ED00014, which contains preferences of 5000 people on various kinds of sushi [59].

For each data set, we generate profiles of 1000 votes by randomly selecting preference rankings from the dataset.

The first thing we show is the percentage of profiles where DCRC is able to change the winner. Figure 3.1 reports the results using the sushi dataset, with 10 voters, $|C| = 5$ and $2 \leqslant |A| \leqslant 5$. The x axis has the value of k in k-approval, which varies from 1 to 4. The four curves correspond to different sizes of set A. Clearly, the larger k and A, the more controllable the profile is, because there could be more harmful combinations of candidate replacements. The same behaviour can be observed in figure 3.2 that reports the results using the t-shirt data set, with 25 voters, $|C| = 5$ and $2 \leqslant |A| \leqslant 5$. The x axis has the value

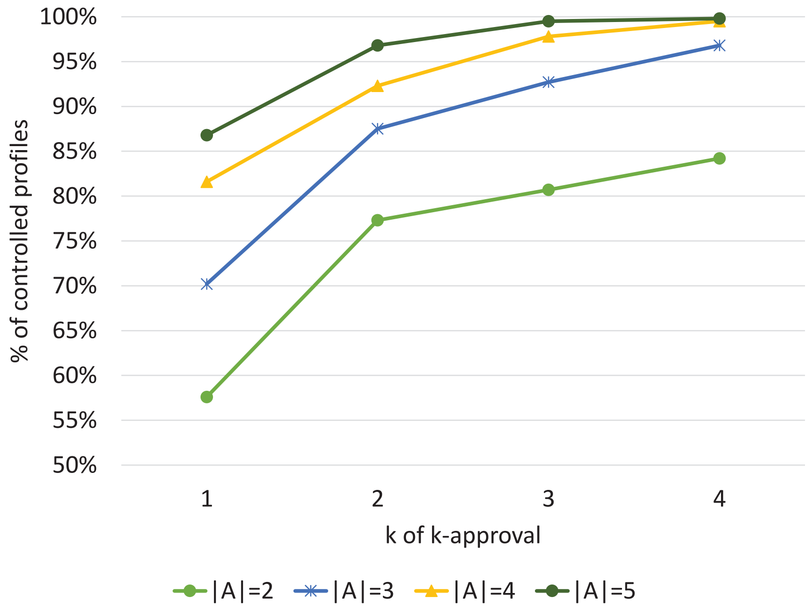

Figure 3.2: T-shirt dataset: fraction of profiles (over 1000) with successful DCRC.

of k in k-approval, which varies from 1 to 4. Even if the data is different we can observe the same trend in both chart: while veto seems to be controllable most of the times, plurality shows some resistance to it.

We then consider the actual difficulty for changing the winner, or for discovering that it cannot be changed, by considering a deterministic algorithm that checks all possible combinations of candidates to be added, and an equal number of candidates to be deleted, starting from combinations with budget (number of replacements) 1 and going up to the maximum size. A lexicographic ordering over candidates is used to decided which delete/add combinations to try first with the same budget size.

Figure 3.3 shows the average fraction of combinations tested, over all possible add/delete combinations, when using 1000 profiles from the sushi dataset, with 10 voters, $|C| = 5$ and $2 \leqslant |A| \leqslant 5$. The x axis has the value of k in k-approval, which varies from 1 to 4. Figure 3.4 shows the same information but against the t-shirt dataset with 25 votes. We are interested in the main trend of these charts. What is really interesting is once again that the larger are k and $|A|$, the smaller is the computational effort of this algorithm.

We also considered a probabilistic algorithm that the chair of the election could use to change the winner by replacing candidates. Such an algorithm consists of picking an add/delete combination randomly (over all possible combinations), and checking whether the winner changes. From the experimental data, we count the percentage of profiles where

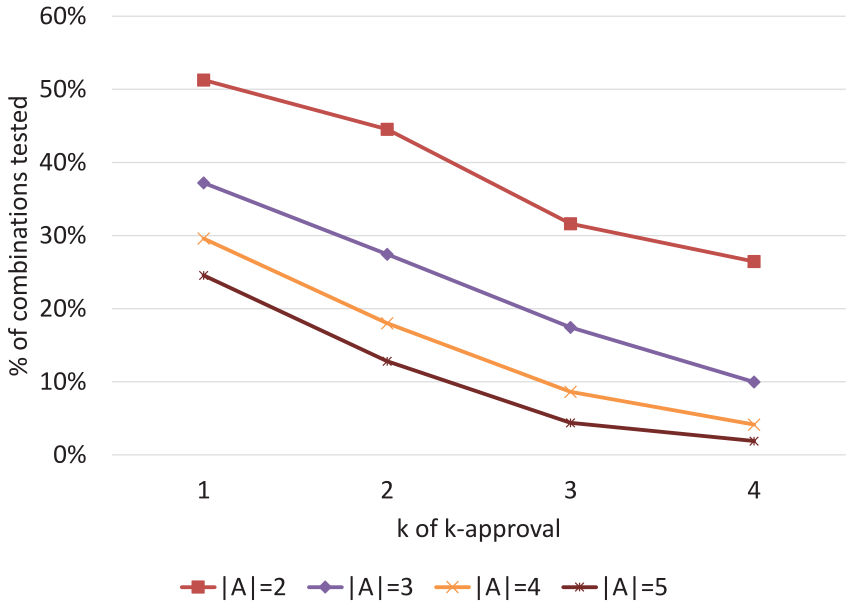

Figure 3.3: Deterministic algorithm on sushi dataset: average percentage of used combinations.

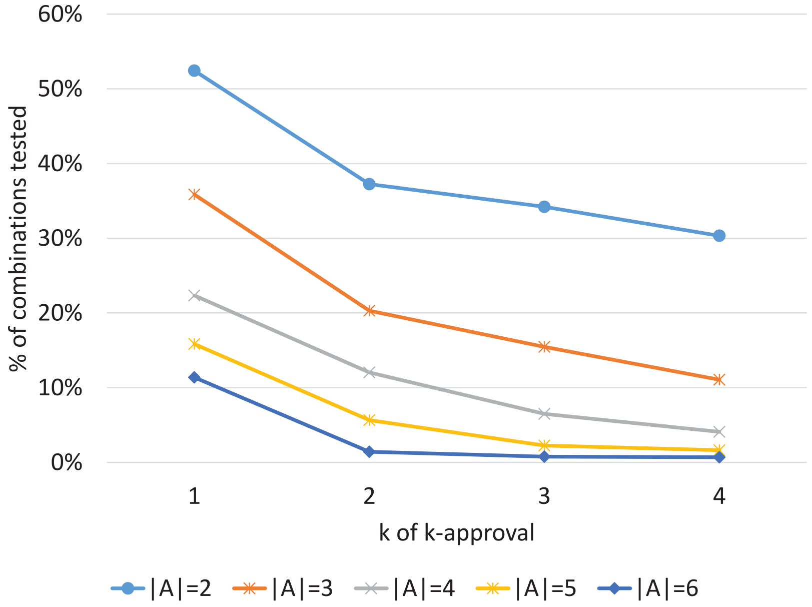

Figure 3.4: Deterministic algorithm on t-shirt dataset: average percentage of tried combinations.

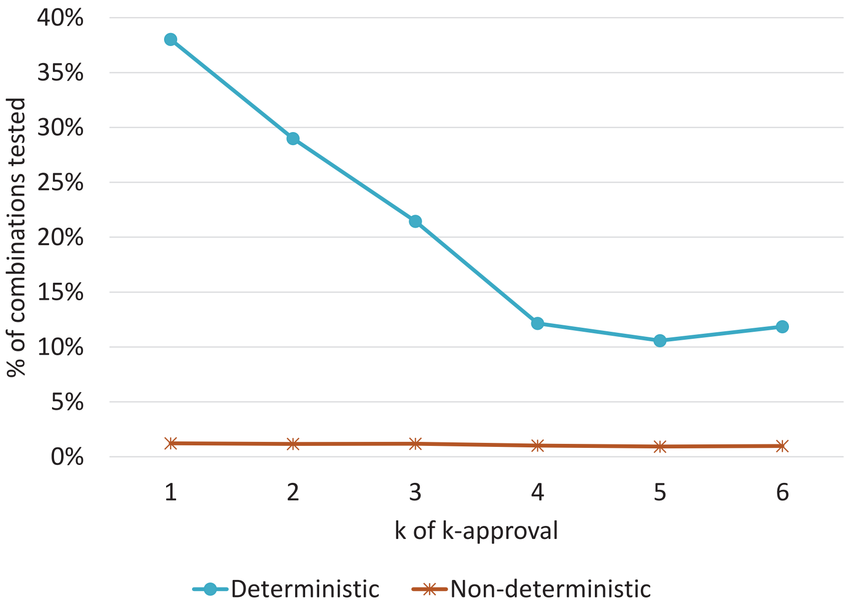

Figure 3.5: Deterministic and probabilistic algorithm comparison on the sushi dataset.

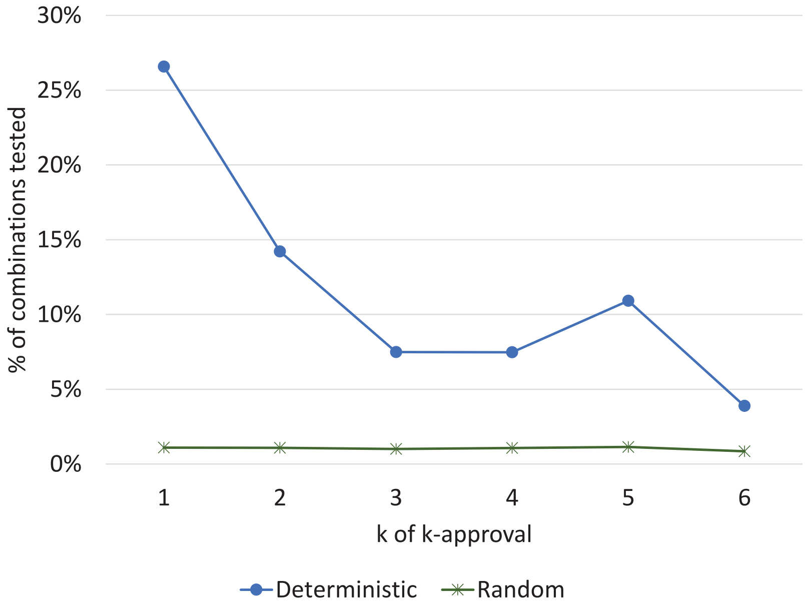

Figure 3.6: Deterministic and probabilistic algorithm comparison on the t-shirt dataset.

the winner changes and we use this as the probability of success of this approach. If p is the probability that picking one profile is enough to change the winner, it is easy to see that $1/p$ is the expected number of profiles to be picked up before changing the winner. We therefore show this $1/p$ number as a measure of how many combinations should be tested by this probabilistic algorithm before changing the result (or discovering that it cannot change).

Figure 3.5 compares the difficulty of the DCRC problem as measured in Fig.2 to this measure of the difficult of DCRC via the probabilistic algorithm. We used the sushi dataset, with 10 voters, $|C| = 7$ and $|A| = 3$. The x axis has the value of k in k-approval, which varies from 1 to 6, while the y axis shows the percentage of add/delete combinations that the algorithm tries before stopping. Figure 3.6 shows the data collected using the t-shirt data set, with 25 voters, $|C| = 7$ and $|A| = 3$. It can be seen that the probabilistic algorithm appears to be more efficient, since it always needs to try a smaller number of combinations.

We also compared the power of replacing candidates with respect to just adding or deleting candidates. Figure 3.7 and Figure 3.8 show the percentage of profiles where the winner changes when the chair uses the replacement control. The bar histograms show the percentage of profiles where the replacement control succeeds and compare this performance with the percentage of profiles where only adding candidates and deleting candidates succeed. We used the sushi dataset, with 10 voters, $|C| = 5$ and $|A|$ varies over the x axis from 1 to 4. The different charts show that increasing the number of candidates that the chair can use the percentage of profiles where the strategic actions succeed increases.

We compared also the power of the single control actions with the replacement control. Figure 3.9 shows a stacked bar histogram for plurality and veto. Each bar represents the percentage of profiles controllable using the replacement control action. The same profiles could be controlled using also different kind of control actions or they can be controlled using only replacement control. For each bar, each different area represents the percentage of profiles which is controllable using only RC but not other forms of control, the percentage of profiles which is controllable using only AC, and so forth also for only DC. We also count the percentage of profile where a combination of AC and DC changes the outcome. Figure 3.9 reports results over the t-shirt dataset, with 10 voters, $|C| = 5$ and $|A|$ varies over the x axis from 1 to 4

It can be seen that RC improves the vulnerability of the voting rule since the number of controllable profiles increases by about 9%, this is a significant increase in controllability compared to AC or DC alone that is not reported in this chart and which is around 0,3%, thus making the voting rule much more vulnerable to this kind of control action.

Figure 3.9 shows data about the same experiment but using the t-shirt dataset. What is

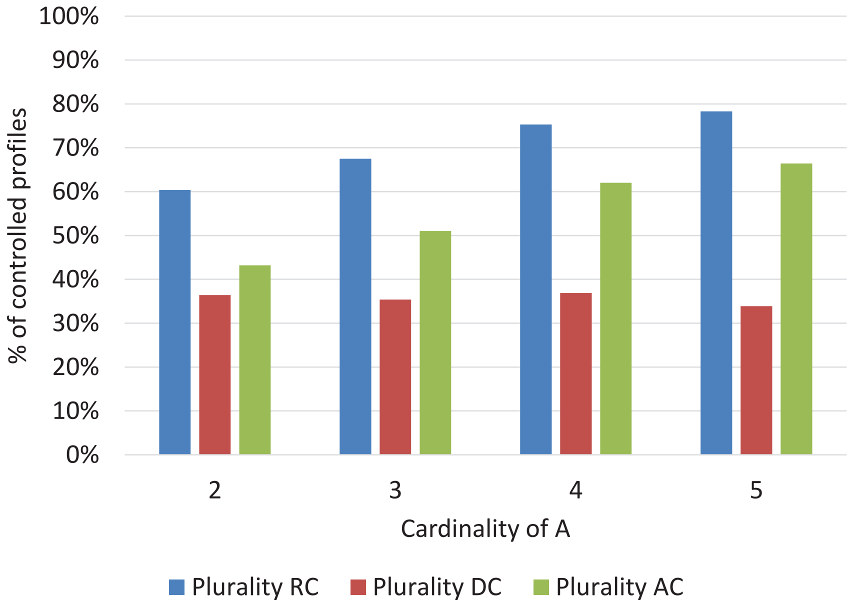

Figure 3.7: Deterministic algorithm on sushi dataset using plurality: RC compared to AC and DC.

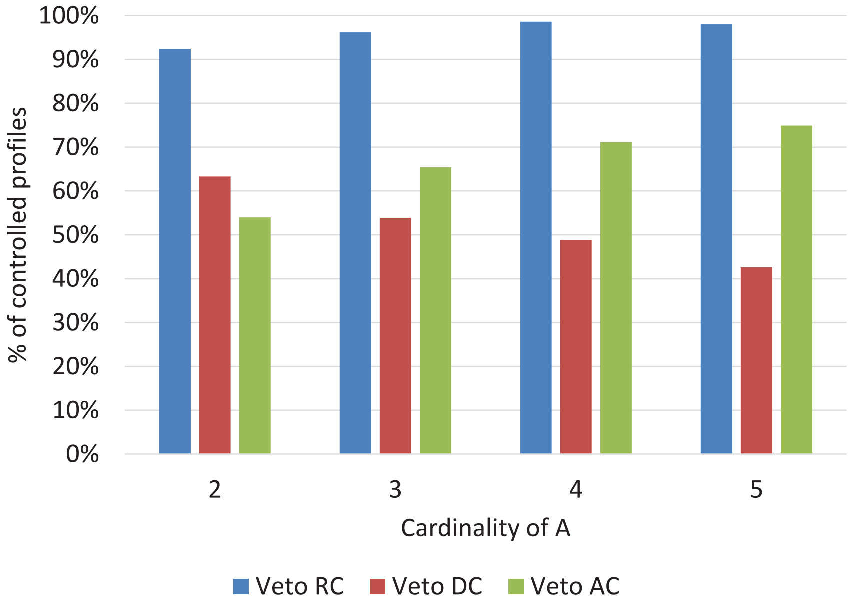

Figure 3.8: Deterministic algorithm on sushi dataset using veto: RC compared to AC and DC.

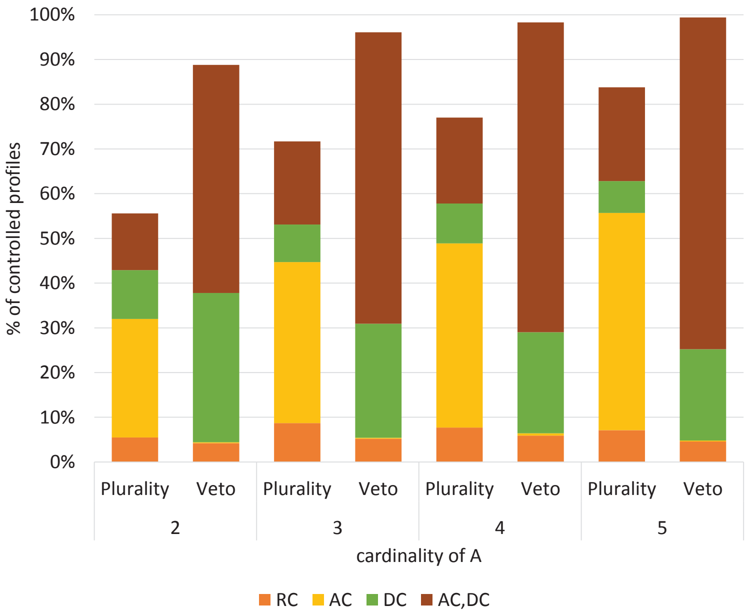

Figure 3.9: Deterministic algorithm: RC compared to AC, DC, and AC+DC.

interesting in this chart is that the structure of the preferences made veto almost resistant to AC only but the voting rule shows the same trend about the vulnerability to RC. Once again RC improves the vulnerability of the voting rule since the number of controllable profiles increases by about 7%, this is a significant increase in controllability compared to AC or DC alone that is not reported in this chart and which is around 0,2%, thus making the voting rule much more vulnerable to this kind of control action.

We also run some experiments using Borda. Theorem 3.4.17 shows that Borda is vulnerable to DCRC. Surprisingly, the deterministic algorithm for checking whether the winner can be changed by replacing candidates needs to test many combinations, as shown in Figure 3.10. Also, the number of profiles where the control succeeds decreases when the cardinality of C increases, as shown in Figure 3.11.

These results suggest that, even if the worst case theoretical analysis tells us that a voting rule is vulnerable to a certain control action, or resistant, the conditions that make it susceptible to the control action could be difficult to find in real-world scenarios, and this could sometimes lead to a reverse situation in practice. Veto, for instance, is resistant to DCRC, but in practice it is very easy to control and this can be done in almost all the profiles. On the other hand, Borda is vulnerable to DCRC, but in practice, when the size of the profile grows, it is unlikely to find a combination that changes the winner.

We also performed experiments with the data collected using the AGH course selection

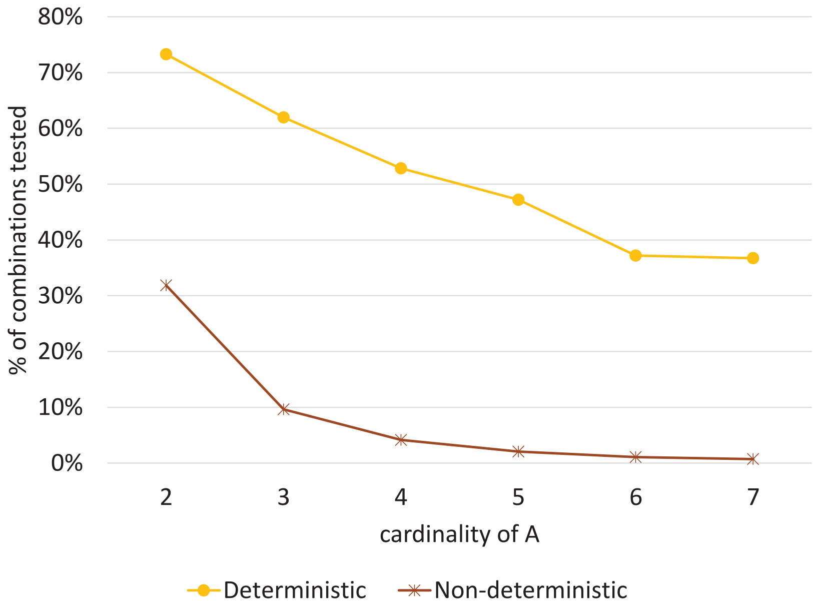

Figure 3.10: Borda deterministic and probabilistic algorithm: comparison.

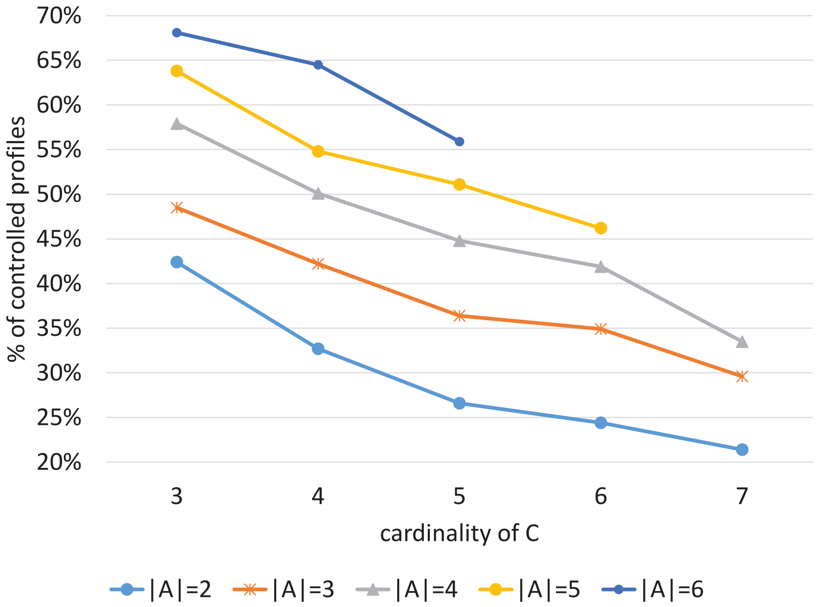

Figure 3.11: Borda: percentage of profiles (over 1000) with successful DCRC.

dataset. However, we do not report them here since they show the same trends as the ones of the other datasets.

4. Iterative Voting

In the previous chapter we considered one-shot elections where manipulation and control are usually considered malicious actions. A voting rule is used to decide which decision to take, mapping the agents' preferences over the possible candidate decisions into a winning decision for the collection of agents. In these kind of scenarios, it may be desirable that agents do not have any incentive to act strategically. Indeed, manipulation and control are usually seen as a bad behavior from an agent, to be avoided or at least to be made computationally difficult to accomplish. We know that every reasonable voting rule is manipulable when no domain restriction is imposed on the agents' preferences [49, 92]. Following this finding, a considerable amount of work has been spent on devising conditions to avoid manipulation from the perspective of the designer of an election. For instance, one can devise restrictive conditions on the preference profiles that can be expressed, or study computational barriers that make the calculation of manipulation strategies too hard for the agents to be performed [9, 44].

In this chapter we consider a different setting, in which instead manipulation is allowed in a fair way. In a first step, we let agents express their preferences over the set of possible decisions, and a voting rule selects the current winner as in the standard case. At this point we give to one agent at a time the possibility to manipulate, i.e., to change her preferences, if by doing so the result changes in her favor. The process repeats with a new agent manipulating until we eventually reach a convergent state, i.e., a profile where no single agent can get a better result by manipulating. This process is called iterative voting, a topic which is getting considerable attention recently in the literature on multiagent systems

[2, 74, 65, 87, 97]. In this scenario, manipulation can be seen as a way to reach a more informed decision, to give every agent a chance to vote strategically in a fair way and to account for inter-agent influence over time.

There are two prototypical situations in which iterative manipulation takes place. The first example is represented by the response of an electorate to a series of information polls about the result of a political election. At each step individuals may realize that their favorite candidate does not have chances to win and report a different preference in the subsequent poll. The second example is Doodle,[1] a very popular on-line system to select a time slot for a meeting by considering the preferences of the participants. In Doodle, each participant can approve as many time slots as she wants, and the winning time slot is the one with the largest number of approvals. At any point, each participant can modify her vote in order to get a better result, and this can go on for several steps.

Iterative voting has been the subject of numerous publications in recent years. Previous work has focused on iterating the plurality rule [74], on the problem of convergence for several voting rules [65], and on the convergence of plurality decisions between multiple agents [2]. If the agents are allowed to manipulate in any way they want (i.e., to provide what is usually called their best response to the current profile), then the iterative process is not guaranteed to converge for most voting rules. Therefore, a crucial problem in the development of iterative voting processes that can be used in practice is the definition of simple manipulation strategies that are able to guarantee convergence. Such strategic actions should not only guarantee convergence, but also be easy to accomplish by the manipulating agent. In fact, contrarily to what we aim for in classical voting scenarios, here we do not want manipulation to be computationally difficult to achieve.

In this chapter we introduce two restricted manipulation moves within the scenario of iterative voting and we analyze some of their theoretical and practical properties. Both manipulation moves we consider are polynomial to compute and require little information to be used. A summary of our results follows:

Convergence. Iterative processes defined using our two proposed manipulation strategies converge for all voting rules we consider, except for single transferable vote (STV) for which we only have experimental evidence of convergence.

Theoretical evaluation. Iterative voting processes define new voting rules, which compute in one stage the outcome of the iteration. We analyze the axiomatic properties as well as the computational complexity of these new voting rules. Moreover, we show that the winner of an iterative process cannot be Pareto dominated by the winner of the initial election.

Experimental evaluation. For voting rules that are not Condorcet consistent, we test

[1] http://doodle.com

whether their Condorcet efficiency (that is, the probability to elect the Condorcet winner) improves by allowing individuals to manipulate in an iterative way. We also tested the evolution of a second parameter, the Borda score of the winner in the truthful profile. Our results show that, with the exception of the Borda rule, both parameters never decrease in iteration, and a significant increase can be observed when the number of candidates is higher than that of voters, as it is the case in a typical Doodle poll. We perform experiments using the impartial culture assumption, a more realistic distribution of preferences called the urn model [14], and real-world datasets from `preflib.org` [72].

There are three possible interpretations and potential applications of our setting. First, iterative voting can be viewed as a description of strategic behavior in real-world electoral situations, and manipulation strategies can therefore be considered as best-response strategies of agents with limited computing capabilities and little access to information about the preferences of the other individuals (descriptive interpretation). Our results show that manipulation in iterated elections, such as repeated opinion polls, does not influence the result in a negative way but may actually lead to the election of candidates with better properties.

Second, restricted manipulation moves may be imposed on agents participating in iterated decision systems such as a Doodle poll (normative interpretation). The manipulation strategies we propose in this chapter all guarantee the convergence of the iterative process and, therefore, that a decision will eventually be reached. Moreover, they are easy to compute and do not require the agents to submit their full preference order, which may be exponential in combinatorial domains.

The third interpretation is that iterative voting defines novel voting rules which compute in one stage the outcome of the iterative process, starting from the full preference orders of the individuals. In this respect, our experiments show that iterative voting rules elect candidates with both a high Borda score and a high Condorcet efficiency, which is surprising given the classical result by Fishburn [45] that no positional scoring rule can be Condorcet-consistent, i.e., have maximal Condorcet efficiency.

The results reported in this chapter are published in several publications. The journal paper first introduces a preliminary study and basic results that are expanded in the subsequent workshop and conference papers.

- Journal papers:
  - Andrea Loreggia. **Iterative voting and multi-mode control in preference aggregation.** *Intelligenza Artificiale 8 (2014) 39–51. DOI 10.3233/IA-140059. IOS Press.*
- Conference papers:

  - Umberto Grandi, Andrea Loreggia, Francesca Rossi, Kristen Brent Venable and Toby Walsh. **Restricted Manipulation in Iterative Voting: Condorcet Efficiency and Borda Score.** *In Proceedings of the 3rd International Conference on Algorithmic Decision Theory (ADT-2013), November 2013*
- Workshop papers:
  - Umberto Grandi, Andrea Loreggia, Francesca Rossi, Kristen Brent Venable and Toby Walsh. **Restricted Manipulation in Iterative Voting: Condorcet Efficiency and Borda Score.** *In Proceedings of the 7th Multidisciplinary Workshop on Advances in Preference Handling (MPREF-2013), August 2013.*
  - Umberto Grandi, Andrea Loreggia, Francesca Rossi, Kristen Brent Venable and Toby Walsh. **Restricted Manipulation in Iterative Voting: Convergence and Condorcet Efficiency.** *In Proceeding of the 1st International Workshop on Strategic Reasoning (SR-2013), 2013.*

4.1 Related work

The framework of iterative voting has first been studied by Meir et al. [74] and by Lev and Rosenschein[65]. In their works, the authors present a game-theoretical formulation of manipulation in a voting problem and study it as a repeated game. They provide conditions for the game to reach a Nash equilibrium varying the voting rule under consideration, the tie-breaking rule and the manipulation strategy used by individuals (although restricted to better response and best response). Additional work on this topic has been done by Reyaneh and Wilson [88], and more recently by Rabinovich et al. [86]. The first paper to consider restricted manipulation strategies in iterated elections is the work of Reijngoud and Endriss [87] on opinion polls, in which a simple manipulation strategy is considered and experiments are performed to evaluate the quality of a winning candidate after a sequence of repeated elections or polls. We review some of their results in Sections 4.2 and 4.3. The more recent work by Obraztsova et al. [82] contains a theoretical analysis of restricted manipulation strategies. Furthermore, the work of Airiau and Endriss [2] studies conditions for convergence in the specific setting of iterated majority voting.

On the experimental side, the work of Thompson et al. [97] focuses on the plurality rule, and studies Nash equilibria in voting games and arrives at similar conclusions as ours on the qualities of winning candidates after iterative best-response manipulation in terms of Condorcet efficiency and Borda score. Finally, iterative voting may have interesting applications in the development of recommender systems which need to collect information in an iterative fashion, such as in the work of Dery et al. [79].

4.2 Background

In this section we recall the basic notions of voting theory that we shall use in this chapter and we present the setting of iterative voting.

4.2.1 Iterative Voting

Voting situations are not immune to strategic reasoning by the voters. Individuals may have incentives to manipulate the election by misreporting their truthful preferences in order to force a candidate they prefer as the winner. The celebrated Gibbard and Satterthwaite Theorem [49, 92] showed that no reasonable voting rule, i.e., one that is not dictatorial and does not rule out *a priori* any candidate, is immune to manipulation when more than 3 candidates are involved.

The setting of *iterative voting* studies repeated voting processes in which individuals are allowed to manipulate one after the other, until convergence is eventually reached. Starting from a profile P^0, a sequence of profiles is then created with the property that at every step P^k only one individual has changed her preference from profile P^{k-1}. It is usually assumed that individuals use a *best response* strategy when choosing what linear order to use to manipulate the election, i.e., they choose the linear order that yields the best possible candidate based on their truthful preference. When the iteration reaches a Nash equilibrium, i.e., a profile in which no individual has an incentive to manipulate the election, we say that the iterative process reaches convergence. Important features of this framework are the sequentiality of the manipulation process (a straightforward observation is that when agents are allowed to manipulate simultaneously then even basic iterative processes do not converge [74]) and the fact that individual agents are myopic, i.e., they have a strategic horizon limited to the current profile under consideration.

Meir et al. [74] and Lev and Rosenschein [65] showed that convergence is rarely guaranteed with most voting rules under consideration, and that this property is highly dependent on the tie-breaking rule used. For instance, while iterative plurality always converges with a linear tie-breaking rule, the iterative version of PSRs other than plurality as well as Maximin do not always converge. The following example shows that the iterative version of the Copeland rule does not converge even if we choose a linear tie-breaking rule.[2]

■ **Example 4.1** Let there be two voters and three candidates, with $a \succ b \succ c$ as tie-breaking order. The initial truthful profile P^0 has $c > b > a$ as the preference of voter 1 and $a > c > b$ for voter 2. Candidate c wins in pairwise comparison with b, and no other candidate win any other pairwise comparison. Thus, the winner using the Copeland rule is c. Voter 2 has

[2] We refer to previous work by Loreggia [69] for an extensive review of results studying the convergence of iterative voting rules.

now an incentive to change her preferences to $a > b > c$, in which case by the tie-breaking order the winner is a, which is preferred by voter 2 in her truthful preference. Now voter 1 is unhappy, and changes her ballot to $b > c > a$ to force candidate b as the winner. This results in an incentive for voter 2 to change her ballot to $a > c > b$, again forcing a as winner. Finally, voter 1 changes again her ballot to $c > b > a$ to obtain c as winner, moving back to the initial profile and creating a cycle of iterated manipulation. The situation is depicted in Figure 4.1, where the voter manipulating is marked on top of the transition arrows.

$$\begin{tikzcd}
\begin{array}{l} v_1 : c > b > a \\ v_2 : a > c > b \\ \hline \text{Winner} = c \end{array} \arrow[r, "v_2"] & \begin{array}{l} v_1 : c > b > a \\ v_2 : a > b > c \\ \hline \text{Winner = a} \end{array} \arrow[d, "v_1"] \\
\begin{array}{l} v_1 : b > c > a \\ v_2 : a > c > b \\ \hline \text{Winner} = a \end{array} \arrow[u, "v_1"] & \begin{array}{l} v_1 : b > c > a \\ v_2 : a > b > c \\ \hline \text{Winner} = b \end{array} \arrow[l, "v_2"]
\end{tikzcd}$$

Figure 4.1: Copeland does not converge with best response manipulation.

■

4.3 Restricted Iterative Voting

In this section we present the setting of restricted iterative voting, in which individuals are allowed to manipulate one at a time by using a restricted set of strategic moves. We introduce two notions of manipulation strategies which are easy to compute from the point of view of the agents, and we show the convergence of the associated iterative process (with the notable exception of STV). We also show that the winner of the iterated voting process after convergence can never be a dominated candidate.

4.3.1 Restricted Iterative Voting Processes

As it is done in the case of iterative voting [74, 65], we consider a sequence of repeated elections with individuals manipulating one at a time at each step. The iteration process starts at a profile which we consider as *truthful*, i.e., every individual reports her real preference. We regard this assumption as realistic, given that in the first stage individuals do not have any information regarding the preferences of the other voters and that a chance to manipulate will be given to them in eventual iteration of the process. In Section 4.5 we perform an initial study of the manipulation of the entire iterative process, in which the initial state may be non-truthful.

Let us first introduce some notation. Let *inf* be a function that associates with a profile of preferences P a set of profiles $inf(P)$, denoting the *information set* available to an agent in profile P. For instance, $inf(P) = P$ in case the agent is fully aware of the profile, and $inf(P) = \{P \mid F(P) = c \text{ and } \forall P_i \in P, P_i = \text{own ballot}\}$ if the agent is only aware of the winner of the current election under F and her own ballot. At each stage of iteration we assume that agents follow a restricted set of strategic moves:

> **Definition 4.3.1** A *manipulation strategy* is a function M, which takes as input a profile of preferences $P^0 = (P_1^0, \ldots, P_n^0)$, an individual $i \in V$, a subset of profiles $inf(P)$, and a voting rule F, and outputs a linear order $M(P^0, i, inf(P), F)$ which identifies the potentially non-truthful preference that will be reported by agent i at profile P knowing that rule F is used. Furthermore, we require that the manipulator with truthful preferences P^0 prefers the new winner $F(M(P^0, i, inf(P), F), P_{-i})$ to the current one given by $F(P)$.

If $M(P^0, i, inf(P), F) = P_i$ we say that individual i in profile P does not have incentives to manipulate. The notion of best response used in iterative voting can be modeled as the following manipulation strategy (assuming that $inf(P) = P$):

$$M_{\text{best}}(P^0, i, P, F) = \text{argmax}^{P_i^0}_{\{P_i' \in \mathcal{L}(C)\}} F(P_i', P_{-i})$$

where $\mathcal{L}(C)$ is the set of all linear orders over C, the maximization is computed following the truthful order P_i^0, and P_{-i} is the profile consisting of all individual preferences in P except for that of individual i. If the maximization is not a singleton then a unique linear order is chosen according to a predefined tie-breaking rule.

We also assume that the iterative process follows a turn function, which identifies the individual that is allowed to manipulate at every step of the iteration:

> **Definition 4.3.2** A *scheduler* or *turn function* is a function τ, which given a sequence of profiles $P^0, \ldots, P^k$ outputs an individual $i \in V$ who is allowed to manipulate at stage k.

The easiest example of a scheduler is the *sequential scheduler* in which voters are manipulating following the order in which they are given. Another example is the *fair scheduler* which associates with each voter i a dissatisfaction index $d_i(k)$ defined as follows: starting from 0, $d_i(k)$ increases by one point for each iteration step in which i has an incentive to manipulate but is not allowed to do so by the turn function. At iteration step k the individual that has the highest dissatisfaction index is allowed to move (in the first step, and in case of ties, the scheduler follows the initial order in which voters are given). A similar notion of scheduler has been introduced by Apt and Simon [3] in weakly acyclic games, and is of crucial importance in showing the convergence of iterative processes.

We can now give the following definition:

Definition 4.3.3 Given a voting rule F, a manipulation strategy M and a turn function τ, a *restricted iterative voting process* is a sequence of profiles $P^0, \dots, P^k$ such that, for all steps $j \leqslant k$, profile P^{j+1} is obtained from profile P^j by having individual $\tau(P^0, \dots, P^i)$ changing her ballot following manipulation strategy M. A restricted iterative process *converges* if there exists a k_0 such that at P^{k_0} no individual has incentive to manipulate.

The current winner can be computed at every stage of the iterative process by applying F to the profile P^k. The Algorithm 5 can be used to compute the final winner in case the restricted iterative process converges:

Algorithm 5 Computing winner

```
Input: Profile $P^0$, voting rule $F$, manipulation strategy $M$, scheduler $\tau$
Output: A winning candidate in $C$ (in case of convergence)
  function COMPUTEWINNER($P^0, i, P, F$)
      $k \leftarrow 0$
      $winner \leftarrow F(P^k)$
      while $has\_incentive(M, P^k)$ do
          $manip \leftarrow \tau(P^0, \dots, P^k)$
          $k \leftarrow k+1$
          $P^k \leftarrow (M(P^0, manip, inf(P^{k-1}), F), P_{-\text{manip}})$
          $winner \leftarrow F(P^k)$
      return $winner$
```

The procedure $has\_incentive(M, P)$ checks that at least one individual has an incentive to manipulate using M at profile P, i.e., that we did not reach convergence. Recall that an individual does not have incentives to manipulate in profile P if $M(P^0, i, P, F) = P_i$, and that P_{-i} is the profile composed of all individual preferences except for i.

4.3.2 Second-Chance and Best-Upgrade

Given the setting of restricted iterative voting, we are interested in devising simple manipulation strategies which guarantee convergence. Previous work by Reijngoud and Endriss [87] proposed a manipulation strategy which can be applied only once by the voters and represents a good starting point for our analysis. Let P^k be the current profile at step k and P^0 be the initial truthful profile:

k-pragmatist (k-PRA) [87]: the manipulator i moves to the top of her reported ballot the most preferred candidate following P_i^0 among those that scored in the top k positions in P^k.[3]

[3] Note that all voting rules considered in this chapter can be easily extended to output a ranking of the candidates rather than just a single winner, by using the score of each candidate to construct an ordering.

Under the *k-pragmatist* restriction, voters are allowed to move to the top of their reported ballot the individual which they prefer amongst the top k candidates ranked by F. We propose the two following definitions:

second-chance (SC): the manipulator i moves the second-best candidate in P_i^0 to the top of her reported ballot P_i^{k+1}, unless the current winner $w = F(P^k)$ is already her best or second-best candidate in P_i^0.

SC allows only a very simple move: switch the first and second candidate in the truthful preference of the manipulator's ballot unless she is already satisfied, i.e., in case the current winner is ranked first or second in her truthful ballot. It represents a simplistic manipulation strategy to obtain baseline results in the setting of restricted iterative voting.

best-upgrade (BU): the manipulator i moves the most preferred candidate in P_i^0 which is above $w = F(P^k)$ in P_i^k to the top of her reported ballot P_i^{k+1}, among those that can become the new winner of the election.

BU instead allows an agent to swap the top candidate with a less preferred one, if this is currently ranked above the current winner. The Algorithm 6 can be used to compute $BU(P^0, i, inf(P), F)$:

Algorithm 6 Computing BU

```
Input: Truthful profile $P^0$, individual $i$, profile $P$, voting rule $F$
Output: A linear order over $C$
  function COMPUTEBU($P^0, i, P, F$)
      $original\_order \leftarrow P_i$
      $better\_than\_winner \leftarrow \{c \in C \mid c\ P_i\ \ F(P)\}$
      while $better\_than\_winner \neq \emptyset$ do
          $d \leftarrow \max_{P_i^0} better\_than\_winner$
          $new\_order \leftarrow upgrade\_top(d, P_i)$
          if $F(new\_order, P_{-i}) = d$ then
              return $new\_order$
          else
              $better\_than\_winner \leftarrow better\_than\_winner - \{d\}$
      return $original\_order$
```

With *BU* the manipulator i selects those candidates that she prefers to the current winner in her current ballot at step k; then, starting from the most preferred one in the truthful ballot P_i^0, she tries to put such candidate on the top of her current ballot (*upgrade_top*) and computes the outcome of the election; the first candidate who succeeds in becoming the new winner of the manipulated election is the one chosen for the top position of her reported ballot. As should be clear from the algorithm, the function *inf* must return sufficient information in order for a single agent to perform winner determination: the score of each candidate for scoring rules, the weighted majority graph for Copeland and maximin, and the full profile for STV.

While the choice of these restrictions may at first seems arbitrary, we believe that they represent three basic prototypes of simple manipulation strategies for agents with bounded computational capabilities and limited access to information. Manipulation strategies can be evaluated following three criteria: (i) the convergence of the iterative voting rule associated with the restriction, (ii) the information to be provided to voters for computing their strategy, i.e., the function *inf* depending on the voting rule, and (iii) the computational complexity of computing the manipulation move at every step. Depending on the interpretation at hand different notions of an ideal restriction can be defined. For instance, if we view iterative voting as a normative process in which we impose restricted manipulation strategies on agents in order to guarantee convergence, an ideal restriction always guarantees convergence, requires as little information as possible, and is computationally easy to compute.

SC requires little information to be computed, i.e., only the winner of the current election, and it is also very easy to compute. The amount of information required by *BU* depends on the voting rule used: the candidates' final score in case of scoring rules, the weighted majority graph for Copeland and Maximin. Instead, in the case of STV the full profile is required. From the point of view of the manipulator, *BU* is computationally easy (i.e., polynomial) to perform. We point to Section 4.5.3 for a more detailed analysis of the computational complexity of performing iterative manipulation.

4.3.3 Convergence

In this section we study the convergence of the restricted iterative voting process associated with our proposed manipulation strategies.

Theorem 4.3.1 An iterative process defined using second-chance (SC) converges for every (deterministic) voting rule F and turn function τ.

Proof. The proof of this statement is straightforward from our definitions. The iteration process starts at the truthful profile P_0, and each agent is allowed to switch the top candidate with the one in second position only once. We stress the fact that in this chapter we consider only deterministic voting rules, i.e., no randomised procedure is used in their definition, therefore when no individual changes their preferences anymore the result of the voting rule remain invariant and the process converges. ■

Theorem 4.3.2 An iterative process defined using best-upgrade (BU) converges for every turn function τ if F is a PSR, the Copeland rule or the Maximin rule.

Proof. The winner of an election using a PSR, Copeland or Maximin is defined as the candidate maximizing a certain score (or with maximal score and highest rank in the tie-breaking order). Since the maximal score of a candidate is bounded, it is sufficient to show that the score of the winner increases at every iteration step (or, in case the score remains constant that the position of the winner in the tie-breaking order increases) to show that the iterative process converges. Let us start with PSR. Recall that the score of a candidate c under a PSR is $\sum_i s_i$ where s_i is the score given by the position of c in ballot P_i. Using BU, the manipulator moves to the top a candidate which lies above the current winner c. Thus, the position – and hence the score – of c remains unchanged, and the new winner must have a strictly higher score (or a better position in the tie-breaking order) than the previous one. Since the maximal score of a candidate is bounded by n times the maximal score that can be given, the process eventually stop. The case of Copeland and Maximin can be solved in a similar fashion: it is sufficient to observe that the relative position of the current winner c with all other candidates (and thus also its score) remain unchanged when ballots are manipulated using BU. Thus, the Copeland score and the Maximin score of a new winner must be higher than that of c (or the new winner must be placed higher in the tie-breaking order). ∎

This proof generalizes to show the convergence of iterative processes using BU for any voting rule where BU does not change the score of the winner and which outputs as winners those candidates maximizing a notion of score.

For the case of STV, we observed experimentally that its iteration always terminates using the fair turn function. However, as shown in the example in Figure 4.2, convergence of STV is not guaranteed if the turn function used is sequential. For space constraints we are omitting the preference symbol $>$ or P between candidates, reading preference from left to right. Let the tie-breaking rule be $e >_C d >_C c >_C b >_C a$, and let voters manipulate using BU. The initial truthful profile is the one on the top left corner.

A closer look at the proofs of Theorem 4.3.1 and Theorem 4.3.2 suggests a bound on the number of iterations of an iterative process defined with our manipulation strategies. Since SC can be applied only once, the iteration process associated to it stops after at most $|V|$ steps for every voting rule. The case of BU is slightly more complex. At every step of the iteration the score of the winner must increase (or she should be placed higher in the tie-breaking order). Thus, in the worst case different winners will touch all possible scores and for each score we will climb up the tie-breaking order until we reach the highest score on the highest tie-breaking position. Since the maximal score of the winner is bounded by a polynomial, so is the number of steps. These observations are summed up in the following statement.

$$\begin{tikzcd}
\begin{array}{l} v_1 : daebc \\ v_2 : bedac \\ v_3 : baedc \\ v_4 : cbeda \\ v_5 : cebda \\ v_6 : adbec \\ \hline \text{Winner} = d \end{array} \arrow[r, "v_2"] & \begin{array}{l} v_1 : daebc \\ v_2 : ebdac \\ v_3 : baedc \\ v_4 : cbeda \\ v_5 : cebda \\ v_6 : adbec \\ \hline \text{Winner} = e \end{array} \arrow[d, "v_1"] \\
\begin{array}{l} v_1 : adebc \\ v_2 : bedac \\ v_3 : baedc \\ v_4 : cbeda \\ v_5 : cebda \\ v_6 : adbec \\ \hline \text{Winner} = b \end{array} \arrow[u, "v_1"] & \begin{array}{l} v_1 : adebc \\ v_2 : ebdac \\ v_3 : baedc \\ v_4 : cbeda \\ v_5 : cebda \\ v_6 : adbec \\ \hline \text{Winner} = a \end{array} \arrow[l, "v_2"']
\end{tikzcd}$$

Figure 4.2: STV with sequential turn function does not converge.

Theorem 4.3.3 An iterative voting process defined using SC will terminate after at most $O(|V|)$ steps. An iterative voting process defined using BU for a PSR, Copeland, or Maximin, converges after at most $O(s \times |C|)$ steps, where s is the maximal score that a candidate can receive in an election.

We conclude by showing that the veto rule does not iterate using our proposed manipulation strategies, and therefore will not be included in our experimental analysis.

Theorem 4.3.4 If $|C| > 3$ the iterative voting processes defined using SC or BU with the veto rule does not iterate, i.e., no agent has incentives to manipulate in the truthful state.

Proof. Recall that the veto rule gives one point to all candidates but the one ranked last in the individual ranking. It is then straightforward to observe that SC cannot change the outcome of the veto rule if there are more than 3 candidates, since the swap occurs only in the top part of the individual preference orders. The same holds for BU: since the candidates that can be upgraded must lie above the current winner of the election, even if the winner is in last position the upgrade will not change the score of any of the candidates, as the veto rule gives one point to all candidates but the last. ■

4.3.4 Quality of the Winner

Let P be a profile of preferences. A candidate z is *Pareto-dominated* by another candidate a in P if it is the case that $a\, P_i\, z$ for all $i \in V$. The following holds:

Theorem 4.3.5 Let F be a PSR and τ be a scheduler. The winner of an iterative voting process defined using SC and BU is never dominated in P^0 by the winner of the initial truthful profile if the latter is higher in the tie-breaking order than the former.

Proof. Let a be the winner of the initial truthful profile, i.e., $a = F(P^0)$, and let z be dominated by a in P^0. Let also a be higher in the tie-breaking order than z. We first deal with the more difficult case of BU and we show by induction that at every step k of the iteration z will still be dominated by a. Given that a PSR cannot elect a Pareto-dominated candidate, we conclude that $F(P^k) \neq z$ and thus that z cannot be the winner of the iterated voting process (recall that by Theorem 4.3.1 the iteration converges under our assumptions).

Let us write $a \succ^k z$ when a Pareto-dominates z in P^k and $a >_{\mathscr{X}} z$. The base inductive case when $k = 0$ is true since $a \succ^0 z$ by assumption. Let therefore P^k be a profile such that $a \succ^k z$ and let $F(P^k) = c$. If $c\, P_i\, z$ for all $i \in V$, in particular if $c = a$, then z cannot be upgraded by BU and will therefore still be dominated by a in P^{k+1}. Hence we can assume that there exists an individual j for which $a\, P_j^k\, z\, P_j^k\, c$, and that j is allowed to manipulate at stage k by the turn function. Since z is above the current winner, individual j is allowed to upgrade z to the top of the ranking provided it becomes the new winner of the election. We show that if this is the case then also upgrading a to the top of the ranking would lead a to be the winner of the manipulated election. Since BU upgrades the best candidate following P^0, and $a \succ^0 z$ by our assumptions, z will not be upgraded and we can conclude that $a \succ^{k+1} z$. Let $s'(z)$ be the score of candidate z if upgraded to the top position by individual j in profile P^k. We assume that z becomes the new winner of the election, and thus that $s'(z) > s(c)$, where $s(c)$ is the score of c in P^k, or in case of equality that z sits higher in the tie-breaking order than c. Let now $s''(a)$ be the score of a if it is upgraded to the top position by j in P^k instead of z. We show that $s'(a) \geqslant s'(z)$ and thus that a can also become the new winner of the election (recall that in case the two scores are equal, a sits higher than z in the tie-breaking order by assumption). Upgrading a or z gives the same score to these two candidates, i.e., the maximal score given by F. All the individuals other than the manipulator j keep their rankings as in P^k, and since $a \succ^k z$ then a gets a higher score than z by all individuals. Thus, a gets a higher or equal score than z if put to the top of the ranking, and we can conclude that $a \succ^{k+1} z$.

In case the iterative voting process is defined using SC the proof is easier. Let P^0 be the initial truthful profile, let $a = F(P^0)$ and let $s^0(a)$ be the score of a in P^0. Let z be

$$\begin{array}{c} v_1 : vazcw \\ v_2 : wcazv \\ v_3 : azcvw \\ \hline \text{Winner} = a \end{array} \xrightarrow{v_2} \begin{array}{c} v_1 : vazcw \\ v_2 : cwazv \\ v_3 : azcvw \\ \hline \text{Winner} = c \end{array} \xrightarrow{v_3} \begin{array}{c} v_1 : vazcw \\ v_3 : cwazv \\ v_3 : zacvw \\ \hline \text{Winner} = z \end{array}$$

Figure 4.3: A dominated candidate elected by PSR with scores (2,1,1,1,0).

dominated by a in P^0, and let $a >_{\mathscr{X}} z$. To show that z cannot be the winner of the iterated election we will show that the score of z at every step k is smaller than the initial score of a, i.e., that $s^k(z) \leqslant s^0(a)$. To see this it is sufficient to observe that manipulation using SC can only increase the score of z by switching it with a in the first position. Thus, the best possible sequence of manipulation will bring z to the position of a, giving z at most the score of a in the initial profile. Since the score of a new winner in the iterated election must have a bigger score than that of a or be ranked higher by the tie-breaking order, we conclude that z cannot become the winner of the iterative process. ■

Observe that if we relax the assumptions of Theorem 4.3.5 by allowing the winner of the iterated process to sit higher than the initial winner in the tie-breaking order, then the proof ceases to hold. This is shown by the following example.

▪ **Example 4.2** Three voters have truthful preferences as in the first profile of Figure 4.3, where preferences are read from left to right. All individuals prefer candidate a to z, i.e., candidate z is Pareto-dominated by a, but let the tie-breaking order be as follows: $z >_C c >_C a >_C v >_C w$. Assume that F is a PSR with scoring vector $(2,1,1,1,0)$. The winner of the election in the initial profile is therefore a. However, as shown in Figure 4.3, successive manipulations using SC yields z as the final winner of the iterated voting process.

▪

A result analogous to Theorem 4.3.5 can be obtained for the Copeland rule.

Theorem 4.3.6 The winner of an iterative voting process defined using SC and BU with the Copeland rule and any turn function is never dominated in P^0 by the winner of the initial truthful profile, provided the former is higher in the tie-breaking order than the latter.

Proof. The structure of the proof for the case of BU is the same as that of Theorem 4.3.5. The inductive step is different, since here we have to show that if the dominated candidate z becomes the new winner of the election by being upgraded to the top position, i.e., z wins against more candidates in pairwise comparisons, then also a can win if upgraded to top position. Let therefore N'_z be the Copeland score of z if upgraded by BU. We want to show that the analogous score $N'_a \geqslant N'_z$ to conclude that $a \succ^{k+1} z$ and prove the inductive step. Since $a \succ^k z$ we know that each pairwise comparison won by z in P^k is also won by a. Now if z is upgraded by j to the top position then z may increase its Copeland score by winning against some candidates that lied above him in P^k. But the same increase will be given to a if it is put to the top position, thus granting him a higher or equal Copeland score (recall that a is ranked higher than z in the tie-breaking order).

Let us now consider manipulation strategy SC. We prove that if z is dominated by candidate a, and $a >_{\mathscr{X}} z$, then z cannot win in any step of the iterative process. The only way in which SC can favor candidate z is by having z win against a in pairwise comparison. Under our assumptions this can happen in case a majority of individuals rank a and z respectively in first and second place in the truthful profile, and these individuals put z on top in subsequent iterations using SC. However, in this situation a is a Condorcet winner of the truthful profile, i.e. she is winning against any other candidate in C, and hence remains the winner in all possible iterations (see Theorem 4.5.2). ■

4.4 Experimental Evaluation

In this section we evaluate four manipulation strategies, namely SC, BU and 2 and 3-*pragmatists*, using two parameters to assess the quality of the resulting winner. First, we measure whether the restricted iterative version of a voting rule has a higher Condorcet efficiency than the initial voting rule, i.e., whether the probability that a Condorcet winner (if it exists) gets elected is higher for the iterative rather than non-iterative rule. Second, we observe the variation of the Borda score of the winner, i.e., we compare the average position of the current winner in the initial truthful profile at convergence with the value of the same parameter in the initial profile. We focus on four voting rules: plurality, STV, Borda, 2 and 3-approval. Our findings show that both parameters never decrease by allowing iterated restricted manipulation, and that a substantial increase can be observed in case the number of candidates is higher than the number of voters (e.g., in a Doodle poll). We conclude the section by reporting on some initial experiments with real-world datasets.

4.4.1 Experimental Setting

We generated profiles using the Polya-Eggenberger urn model (see, e.g., [14]). Individual ballots are extracted from an urn initially containing all $m!$ possible ballots, i.e., all linear orders over m candidates, and each time we draw a vote from the urn we put it back with a additional copies of the same vote. In this way we generate profiles with correlated preferences and we control the correlation ratio with the parameter a. In our experiment we tested three different settings: the *impartial culture assumption* (IC) when $a = 0$, the UM10 with 10%-correlation when $a = \frac{m!}{9}$, and the UM50 with 50%-correlation when $a = m!$.

Our results are obtained using a program implemented in Java ver.1.6.0.[4] We model two prototypical examples of iterative voting: in the *electoral simulation* we set the number of candidates to $m = 5$ and the number of voters to $n = 500$ to model situations in which a large population needs to decide on a small set of candidates. In the *Doodle simulation* we set the number of candidates to $m = 25$ and the number of voters to $n = 10$ to model a small group of people deciding over a number of time slots. In both cases we performed additional experiments with similar number of voters and candidates without observing significant variation in the results. In all cases we validated our conclusions using Mc Nemar's test, obtaining results with p consistently below 0.00001. In our experiments we used the fair turn function defined in Section 4.3. We also performed initial experiments using the sequential turn function obtaining similar results to those shown below (with the exception of STV, which does not converge using the sequential turn function).

It is interesting to observe that the higher the correlation in the profile the smaller the number of profiles in which iteration takes place (with the notable exception of the Borda rule). In Figure 4.4 it is shown, for the Doodle simulation, the percentage of profiles in which iteration takes place for the three different correlation ratios considered. In the case of plurality, convergence is reached after an average of 3 steps and a maximal of 9 steps. The figures for the other voting rules are similar.

4.4.2 Condorcet Efficiency

Figures 4.5 and 4.6 compare the four restrictions on manipulation moves with respect to the Condorcet efficiency of the iterative version of the five voting rules under consideration, respectively for the Doodle simulation and the electoral simulation. In both experiments the correlation ratio is set at 10% and we generated 10.000 profiles all with Condorcet winner.

Except for the case of the Borda rule, the Condorcet efficiency of the iterative version of a voting rule always improves with respect to the non-iterative version, and the growth is consistently higher when voters manipulate the election using *BU* rather than *SC*. Let us also stress that while the increase in Condorcet efficiency using *SC* is minimal, it is still surprising that such a simple move can result in a better performance than the original version of the voting rule. The 2-*pragmatist* restriction performs quite well with the plurality rule in both experiments. STV has the highest performance of all voting rules considered thus far with respect to Condorcet efficiency and this performance is amplified by the use of iterated manipulation, resulting in the election of a Condorcet winner in almost 95 percent of the cases. As remarked earlier, we observed convergence in all profiles considered. The increase in Condorcet efficiency is more noticeable in the

[4] The code of the program used for the experiments is made available at the following address: www.math.unipd.it/~loreggia/download.html

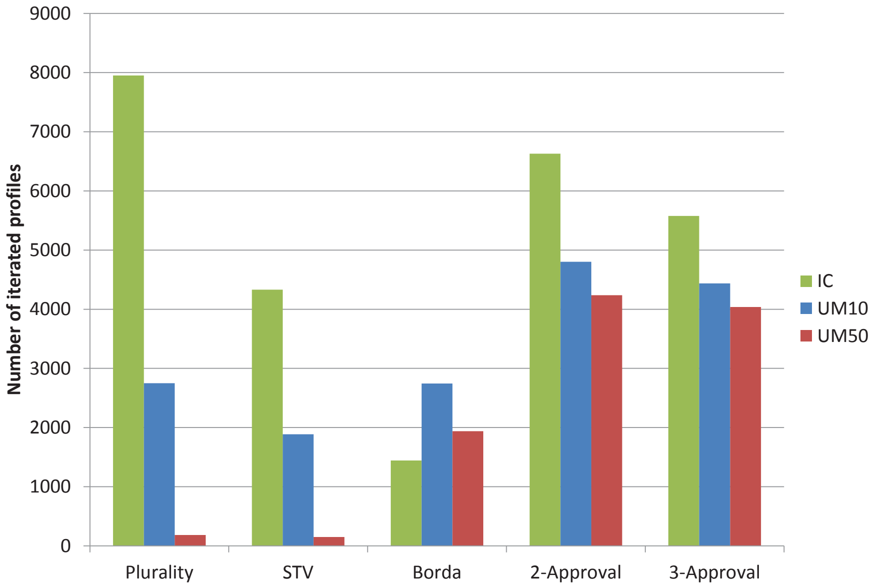

Figure 4.4: Number of profiles with iteration compared to the correlation ratio.

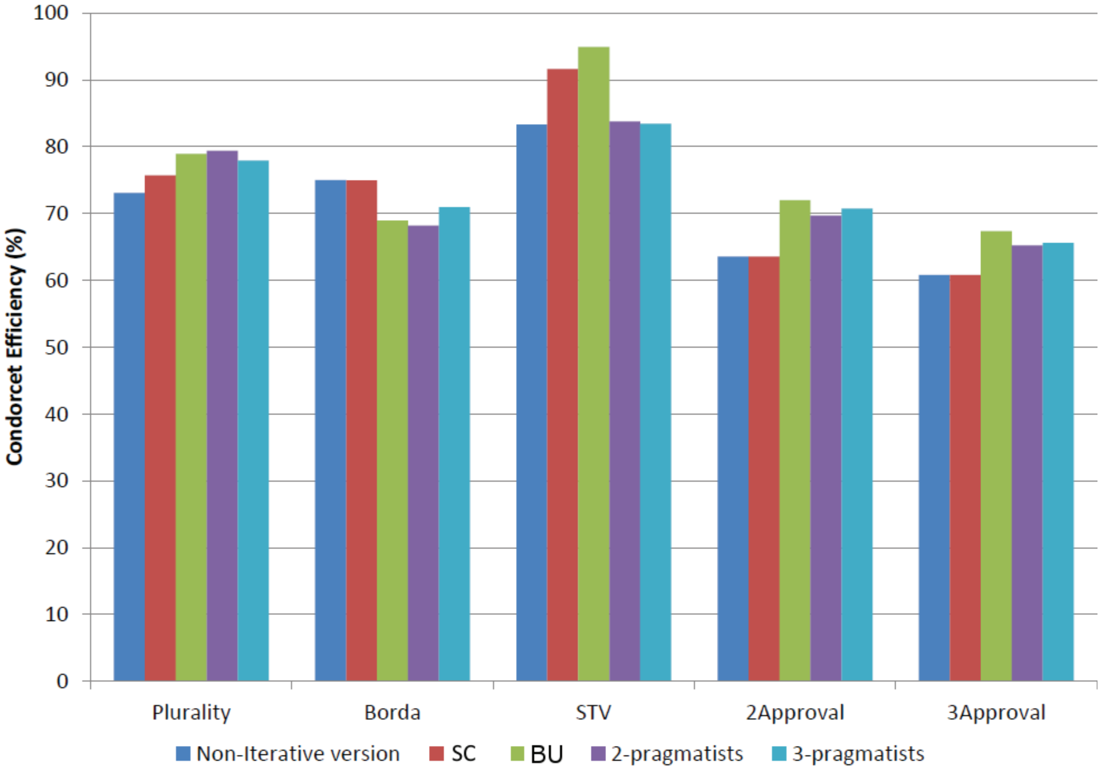

Figure 4.5: Doodle experiment with UM10: Condorcet efficiency.

Doodle simulation rather than in the electoral situation. When the number of individuals is considerably higher than the number of alternatives the iterative process leads to a minimal increase in Condorcet efficiency.

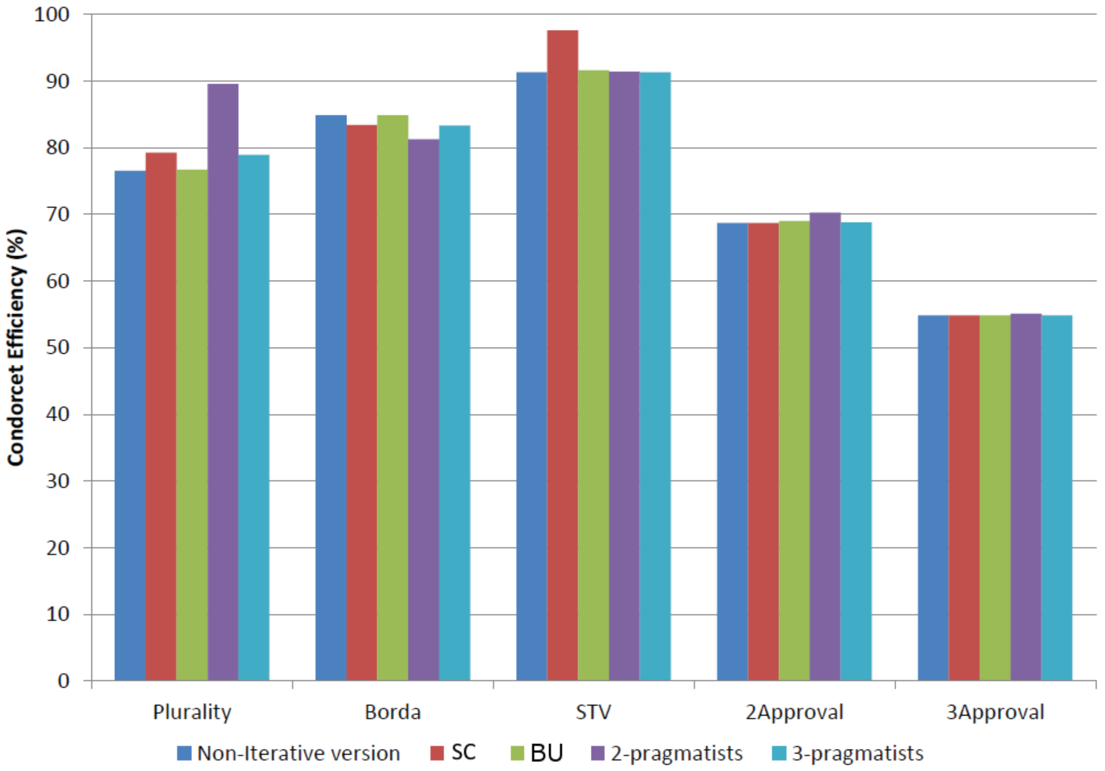

Figure 4.6: Electoral experiment with UM10: Condorcet efficiency.

We also run the same two experiments with different correlation ratios. Using the IC assumption the increase in Condorcet efficiency is more significant, while with the UM50 assumption the results are much less perturbed by iteration. This should not come as a surprise, given that the amount of profiles in which iteration takes place decrease rapidly with the growth of the correlation ratio.

4.4.3 Borda Score

The second parameter we used to assess the performance of restricted manipulation moves is known in the literature as the Borda score. Given a candidate c, let p_i be the position of c in the initial preference b_i^0 of voter i (from bottom to top, i.e., if a candidate is ranked first she gets $m-1$ points, while if she is ranked last she gets 0 points). We compute the Borda score of c as $\sum_{i=1}^{n} p_i$.

For each voting rule and each restriction on the set of manipulation moves we compared the score of the winner of the non-iterative version with that of the winner of the iterative version after convergence over 10.000 profiles. A particular case is that of the Borda rule.

By definition this rule elects those candidates with the highest Borda score, hence we did not evaluate the evolution of this parameter in this case. Our results showed that in both the Doodle and the electoral simulation with UM10 the Borda score increases minimally if we allow for iterated restricted manipulation, resulting in a chart similar to that in Figure 4.6. The best results are in this case obtained by using *BU* and 2-*pragmatists* restrictions with 2 and 3-approval. As in the previous section, by decreasing the correlation of the generated preferences we obtain a more significant increase in the Borda score after iteration.

4.4.4 Real-world Datasets

We performed experiments using data from `preflib.org` [72], a library of preference datasets collected from various sources. In order to mimic the original preference distribution, we generated 10.000 profiles with 5 candidates and 50 voters drawing votes uniformly from three original datasets: the Netflix Prize Data [13], the Skating Data, and the Sushi data [59]. What we observed is that preferences contained in such datasets are quite correlated, with many profiles being composed by almost unanimous orders. The results of the iteration are therefore negligible, as can be seen in Figure 4.7, and iteration takes place in just a handful of profiles (in the order of 5–10 per 10.000 profiles) in accordance with the results obtained for the highly correlated urn model UM50.

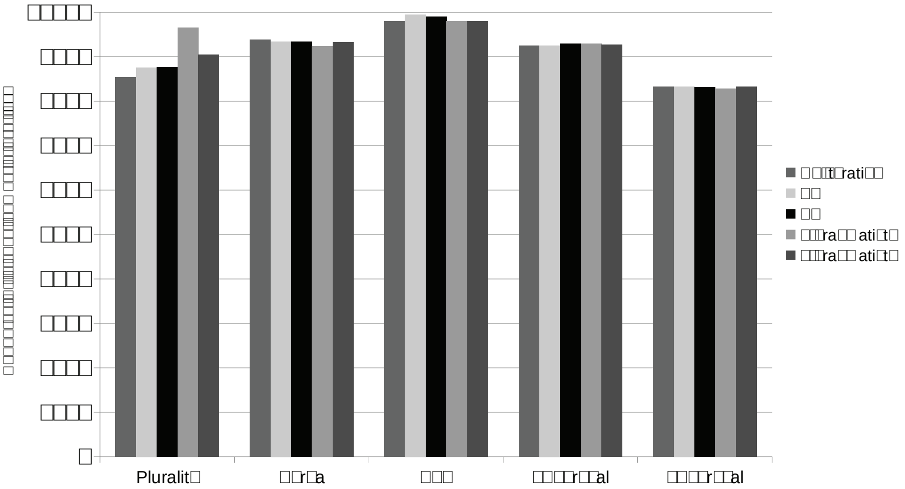

Figure 4.7: Sushi Dataset experiment: Condorcet efficiency

These results suggest that iterative voting is not of practical use with highly correlated preferences, e.g, in situations such as the measuring of objective qualities of movies, sushi or skating performances. As the UM10 Doodle experiment shows the best performance of restricted iterative voting are instead obtained in situations where preferences are less correlated and the group of voters is small with respect of the number of alternatives.

4.5 Iterative Processes as a One-Stage Voting Rule

In this section we view iterative voting processes as a one-stage voting rule, which collects the full preference order from the individual voters and runs the sequence of manipulation moves until reaching convergence. We show that many axiomatic properties that are satisfied by a voting rule are then transferred to its iterative versions, depending on the manipulation strategies used. We also analyze the computational complexity of computing the winner of these novel voting procedures and we conjecture that for some of those rules the problem of strategic manipulation may become computationally hard.

4.5.1 One-Stage Iterative Voting Rules

The following definition presents the one-stage iterative version of a voting rule depending on the different manipulation strategies used:

Definition 4.5.1 Let F be a voting rule, τ a turn function and M a manipulation strategy. F^M associates with every profile P the outcome of the iteration of F using turn function τ and manipulation move in M in case it converges, and symbol $\uparrow$ otherwise.

4.5.2 Axiomatic Properties

Voting rules are traditionally studied using axiomatic properties, and we can inquire whether these properties extend from a voting rule to its iterative version. We refer to Section 2.4 for an explanation of these properties.

We say that a restricted manipulation move M *preserves* a given axiom if whenever a voting rule F satisfies the axiom then also F^M does satisfy it.

Theorem 4.5.1 *SC* and *BU* preserve unanimity for every turn function. *SC* and *BU* preserve neutrality if the turn function is neutral. [a]

[a] I.e., if the turn function is invariant under permutation of alternatives.

Proof. Assume that the iteration process starts at a unanimous profile P in which candidate c is at top position of all individual preferences. If F is unanimous, then $F(P) = c$, and no individual has incentives to manipulate either using SC or BU. Thus, iteration stops at step one and $F^{SC}(P) = c$ and $F^{BU}(P) = c$, satisfying the axiom of unanimity. The proof for anonymity and neutrality is straightforward from our definitions. ■

A further important property of voting rules is *Condorcet consistency*: a rule is Condorcet consistent if it elects a Condorcet winner when it exists. Examples of Condorcet-consistent rules are Copeland and maximin.

Theorem 4.5.2 *SC* and *BU* preserve Condorcet consistency.

Proof. Let c be the Condorcet winner of a profile P. If F is Condorcet-consistent then $F(P) = c$. When individuals manipulate using either SC or BU the relative position of the current winner with all other candidates does not change, since the manipulation only involves candidates that lie above the current winner in the individual preferences. Thus c remains the Condorcet winner in all iteration steps P^k of the iterative process. Since $F^{SC}(P) = F(P)$ and F is Condorcet-consistent, we have that $F^{SC}(P) = c$ and thus F^{SC} is Condorcet consistent. The case of BU is easier: since it is not possible to change the winner of the initial truthful election, i.e., the Condorcet winner, then by the definition of BU there will be no iteration and the iterated voting rule yields the same result as the initial voting rule. ■

4.5.3 Computational Complexity

The complexity of the winner determination problem requires us to determine the time necessary to compute the outcome of a voting rule F on a given input. This problem has been widely investigated for many voting rules, together with the complexity of many other problems related to voting such as manipulation and bribery (see, e.g., [43]). The problem of winner determination should be easy to solve to render a voting rule interesting for practical applications (and this problem is polynomial for all voting rules we considered).

For all these rules for which we have proven convergence in Section 4.3.3 we are able to show that the problem of winner determination remains polynomial also for the iterative version:

Theorem 4.5.3 The winner determination problem is polynomial for iterated PSRs, Copeland and Maximin using restricted manipulation moves SC and BU.

Proof. The proof consists of the following observations: at each iteration step, a manipulation strategy should be used to determine the new profile, and the current winner be computed (recall the algorithms presented in Section 4.3). All manipulation strategies we considered can be computed in polynomial time, and the winner determination problem is also polynomial for PSRs, Copeland and Maximin. Hence the outcome of every iteration step can be computed in polynomial time. We can now conclude by using Theorem 4.3.3 to obtain a polynomial upper bound on the number of iteration steps for both manipulation strategies SC and BU. ■

The problem of the computational complexity of strategic manipulation in iterated voting processes is rather complex, and we leave it as an open question. Consider an iterative voting process, and a voter which is submitting her first vote, which we assume as truthful. If provided with enough information about the preferences of the other individuals, this voter may compute the outcome of the iterated voting rule for each preference order she may submit, and thus behave accordingly manipulating the entire iterative voting process. However, observe that in this case we drop one of the crucial assumptions of iterative voting processes, as individuals are not myopic but act strategically with an arbitrarily long horizon.

All voting rules we considered except for STV are easy to compute, but also to manipulate. A voting procedure becomes attractive when the winner determination problem is polynomial while manipulation is NP-hard (as it is the case for STV). An investigation of the computational complexity of manipulating F^M varying the manipulation strategy M and the voting rule F has the potential of unveiling new similar rules. Further arguments in this direction can be taken from the recent work by Narodytska and Walsh [80], which suggest that adding multiple steps to a voting rule may lead to a significant increase in the computational complexity of manipulation.

5. Sentiment analysis

We live in a world where we communicate more and more on social media, writing text that reflects our opinions and feelings. Being able to formalize such opinions and reason with them can be very useful for a number of practical applications. First, service providers may personalize their offer based on customers opinions. Second, companies may test what products would be better received by potential consumers, and adjust their strategy accordingly. Third, community councils and candidates in political elections may evaluate the reception of their proposals, and focus their attention on the most preferred ones. It comes therefore as no surprise that the extraction of individual opinions from textual expressions, such as tweets, blog posts, or product reviews, has been the subject of a very active area of research in recent years.

Researchers in sentiment analysis and opinion mining [**?** 83] developed a collection of tools in natural language processing (NLP) for the extraction of opinions, sentiments, or attitudes of individuals from their textual expressions. In order to summarize the opinion of all the individuals in a unique indicator, the opinions extracted are then used to define a notion of collective sentiment about the entities under consideration, be they commercial products, policies or candidates.

In this chapter we observe that current sentiment analysis techniques are good enough when we are trying to understand the positive or negative opinion of a set of agents over a single item, but they fall short when we are considering several items. Our claim stems from the observation that, when several items are being compared, the approach taken by sentiment analysis of only focusing on positive or neutral polarities may differ from the approach that computes the most preferred item by making use of comparative preference information. Consider for instance the following situation, in which two candidates Ann and Bob are competing in an election.

■ **Example 5.1** Assume there are a total of 35 people who are expressing their positive or negative attitude on social media about two candidates Ann and Bob: 20 persons are talking positively about Ann, 15 persons are talking negatively about Ann, 30 persons are talking positively about Bob, and 5 persons are talking negatively about Bob. However, what people write on social media is just a textual abstraction of the comparative preferences they have in mind, which in this particular case we assume to be a ranked list of the two candidates. Assume therefore that their preferences are as described in the following table, where candidates to the left are more preferred than candidates to the right, and the bar signals the threshold of positive vs. negative opinions:

<table>
<tr><td>20 voters:</td><td>Ann</td><td>Bob</td><td>|</td><td></td><td></td></tr>
<tr><td>10 voters:</td><td></td><td>Bob</td><td>|</td><td>Ann</td><td></td></tr>
<tr><td>5 voters:</td><td></td><td></td><td>|</td><td>Ann</td><td>Bob</td></tr>
<tr><td>Sentiment analysis</td><td></td><td></td><td></td><td>Bob</td><td></td></tr>
<tr><td>Majority rule</td><td></td><td></td><td></td><td>Ann</td><td></td></tr>
</table>

In the profile described above there are 30 voters that express a positive opinion about Bob, and 20 voters that express a similar opinion about Ann. Hence, sentiment analysis, as well as similar methods based solely on sentiment information, would conclude that Bob is the most popular candidate. However, if we assume that the election will be decided by majority voting, then Ann will be the winner of the election with 25 votes over 10 for Bob, unlike the outcome of sentiment analysis. Observe that the positive/negative opinions expressed by the individuals are consistent with the preferences that will then be revealed at the time of voting. ■

The situation above is a good example of the use of sentiment analysis and preference aggregation for the *prediction* of a real-world event. In this particular case, a prediction of an electoral result is being based on the number of positive opinions extracted from voters. Similar examples can also be devised to point out a problem in situations of *decision-making*: think of Ann and Bob as two products that a firm is considering to promote, and the sentiment and preferences expressed in the table be those extracted from conversations and reviews of its customers. When the firm needs to decide which of the two products to invest in, sentiment analysis and preference aggregation would give two different recommendations.

The first message of this chapter is that all these considerations can be phrased in the framework of preference analysis [90, 98] and voting theory [6]. For instance, sentiment analysis as presented in the example above uses a preference aggregation method called approval voting [18], which is based only on positive or negative opinions expressed over candidates. Text-extracted opinions may present both polarities and preference orderings, and the main contribution of this chapter is to propose a definition of collective sentiment that makes use of both kinds of information.

Building on the classical Borda count (see, e.g., [17]) we define and study a class of voting rules that aggregate both polarities and preference orderings into a collective sentiment, taking into account the incompleteness inherent in text-extracted opinions, where each individual may refer only to some of the items under consideration. We study the behavior of this class of rules from a decision-theoretic perspective. First, we list a number of properties that are desirable in the context of sentiment analysis, and we show that our proposed rules satisfy all such conditions. Second, we perform experiments to quantify the discrepancy between classical sentiment analysis techniques and our proposed rule, and we investigate its behavior with respect to partial information. The results we obtain indicate that our proposed Borda count not only satisfies a list of desirable properties when its outcome is used as a basis for decision-making, but also it is computationally tractable and it behaves well in highly incomplete domains.

To the best of our knowledge this work is the first attempt to apply techniques from preference aggregation and voting theory to sentiment analysis over multiple issues. Related work has focused on sketching a road map for developing sentiment analysis as an alternative to opinion polls for the prediction of electoral results [76], focusing however on the statistical significance of the population studied rather than on the aggregation method used. Preference aggregation techniques have been used with success in other areas of computer science such as human computation and collective annotation of textual corpora [34, 71], and on developing procedures for collective decision making that are able to handle incomplete preferences [85, 101]. A line of work which is similar in spirit to the one proposed in this chapter is the work of Brams and Sanver [16] in social choice theory, albeit for the specific setting of committee decisions and elections. We refer to Section 5.3.3 for a more detailed discussion of this approach. We also acknowledge the work of Garg et al. [48] on opinion pooling, which is however focused on the aggregation of probabilistic opinions.

The work of the thesis has been presented in several international conferences. This is the list of papers related with this work classified for journal, conference or workshop where they were presented.

- Journal papers:
  - Umberto Grandi , Andrea Loreggia, Francesca Rossi, Vijay Saraswat. **A Borda count for collective sentiment analysis.** *Annals of Mathematics and Artificial*

Intelligence - 75 (149): 1-22, 2015.

- Conference papers:
  - Umberto Grandi, Andrea Loreggia, Francesca Rossi, Vijay Saraswat. **From Sentiment Analysis to Preference Aggregation.** *ISAIM 2014 special session on computational social choice.*
- Workshop papers:
  - Umberto Grandi, Andrea Loreggia, Francesca Rossi. **From Sentiment Analysis to Preference Aggregation.** *R.i.C.e.R.c.A: RCRA Incontri E Confronti Workshop of the XIII AI*IA Conference.*

5.1 Background

In this section we present the basic definitions of sentiment analysis. Notions about voting theory and preference aggregation are based on Chapter 2.

5.1.1 Sentiment Analysis

Sentiment analysis and opinion mining [83**?**] is a collection of techniques for the extraction of people's opinions, sentiments, and evaluations from textual expressions. A set of entities or alternatives $\mathscr{X}$ is defined as the sentiment targets, and individual opinions about entities in $\mathscr{X}$ are extracted from a given set of product reviews, blog posts or other sources of textual information.

Formally, two forms of opinions can be identified:

Definition 5.1.1 [67] A *regular opinion* is a tuple (g,s,h,t) where g is the sentiment target, s is the sentiment about the target, h is the opinion holder and t is the opinion time.

Definition 5.1.2 [58] A *comparative opinion* (e_1,e_2,pa,h,t) is a tuple where e_1 and e_2 are two entities that are being compared, pa is the preferred alternative among e_1 and e_2, h is the opinion holder and t the time.

Sentiment targets are also called entities or items, and can be anything such as products, policies or persons. The sentiment s in a regular opinion is usually taken to be a positive, negative or neutral polarity, i.e., an element of $\{+,-,0\}$, although recent developments are directed to a more general setting of graded polarity such as a "five-stars" scale or numerical score [37]. The opinion holder h is the individual who wrote a text expressing sentiment s, and the time t is the moment at which h wrote the text. In this chapter we will not make use of the temporal information, but we refer the reader to Chapter 6 for further discussion on the important role that temporal information may play in the development of principled notions of collective sentiment.

The objective of sentiment analysis is to extract all possible opinion tuples as in Definitions 5.1.1 and 5.1.2. Popular approaches to perform this task use a bag of words extracted from a tagged corpus of positive sentences, and then count in a more or less complex way the presence of such positive words in untagged documents [84]. Machine learning techniques such as naive Bayes approaches and sentiment classifiers built using semi-supervised learning are also widely used for these tasks (see, e.g., [84, 7] for regular opinions and [58, 46] for comparative opinions).

A notion of *collective sentiment* aggregates individuals' opinions into a collective view, and it is usually expressed as a polarity. The most common approaches define the collective sentiment as a positive sentiment if the number of positive opinions about the item outnumbers the number of negative opinions. When more than one item is considered, each textual expression is classified as positive, negative or neutral, and the items with the largest number of positive expressions are declared as the most preferred ones according to the collective opinion (see, e.g., [77, 8, 20]).

5.2 How to model individuals' opinions

Sentiment analysis and preference aggregation take two different approaches in the representation of absolute and comparative preferential information that is extracted from individual data. The aim of this section is to formally define these two approaches, and to propose a novel structure for preference representation that combines sentiment polarity with comparative preferences.

5.2.1 Individual data

We assume to have collected a set of textual expressions $\mathscr{T}_i$ for every individual $i \in \mathscr{I}$, and that exploiting tools from NLP we are able to extract regular opinions expressed by individuals about the entities in a set $\mathscr{X}$ in the form of a score (see Definition 5.1.1), as well as comparative opinions in the form of binary comparisons (see Definition 5.1.2).

Definition 5.2.1 The *individual data* extracted from a set of individual expressions $\mathscr{T}_i$ is a tuple $(\sigma_i, \leqslant_i^P, \leqslant_i^N)$ where:

- $\sigma_i : D_i \to \mathbb{R}$ is a function defined on a subset of entities $D_i \subseteq \mathscr{X}$ representing all regular opinions, i.e., degrees of positive and negative opinions over entities;
- $\leqslant_i^P$ is a preorder with domain $P_i \subseteq \mathscr{X}$, representing the set of positive comparative opinions of individual i;
- $\leqslant_i^N$ is a preorder with domain $N_i \subseteq \mathscr{X}$ representing the set of negative comparative opinions of individual i.

We make the further assumption that the individual data is always coherent, i.e., the sets P_i and N_i are disjoint sets and when entities a and b are in P_i (resp. N_i) then both $\sigma_i(a)$ and $\sigma_i(b)$ are positive numbers (resp. negative), and also that $a \nleqslant_i^P b$ (resp. $a \nleqslant_i^N b$) if $\sigma_i(b) < \sigma_i(a)$. Observe moreover that the sets D_i, P_i and N_i may have non-empty intersection.

■ **Example 5.2** A company wants to evaluate three products of different colors: red (R), green (G) and blue (B). A corpora of textual expressions by three individuals is collected and the individual data extracted is as follows. The first individual has a positive opinion about all three products R, G, B, but the degree of these opinions is slightly different: we extract a score of 5 for product R, a score of 4 for entities G and B, and no pairwise comparison among the products. The second individual has a positive score of 1 about product G while expressing a negative opinion about the other two colors. She also expresses a direct preference of R over B. Finally, the third individual has a neutral opinion about R and B, while she considers alternative G negatively with a score of -4. We can summarize the opinions extracted from the three individuals in the terminology of our Definition 5.2.1:

- Individual 1: $\sigma_1(R) = 5, \sigma_1(G) = \sigma_1(B) = 4$ and $P_1 = N_1 = \emptyset$;
- Individual 2: $\sigma_2(G) = 1$, $P_2 = \emptyset$, and $N_2 = \{R, B\}$ with $B \leqslant_2^N R$;
- Individual 3: $\sigma_3(R) = \sigma_3(B) = 0$, $\sigma_3(G) = -4$, and $P_3 = N_3 = \emptyset$.

■

5.2.2 The sentiment analysis approach

Sentiment analysis (at least in its most common implementation) disregards the intensity of sentiment as well as the comparative opinions, focusing only on the extraction of a positive, negative or neutral polarity from individual expressions.

Definition 5.2.2 Given individual data $(\sigma_i, \leqslant_i^P, \leqslant_i^N)$ extracted from individual expressions $\mathscr{T}_i$, the *pure sentiment data* associated with it is a function $Sent_i : E_i \to \{+, -, 0\}$, where $E_i = D_i \cup P_i \cup N_i$, defined as:

$$Sent_i(c) = \begin{cases} \operatorname{sgn}(\sigma_i(c)) & \text{if } c \in D_i \\ 0 & \text{if } \sigma_i(c) = 0 \\ + & \text{if } c \in P_i \\ - & \text{if } c \in N_i \end{cases}$$

After the information about the individual sentiments have been extracted, the most common approach in the definition of the collective sentiment is to choose the entities with the largest amount of positive opinions, disregarding the number of negative opinions.

■ **Example 5.3** The pure sentiment data associated with Example 5.2 is the following:

- Individual 1: $Sent_1(R) = Sent_1(G) = Sent_1(B) = +$
- Individual 2: $Sent_2(G) = +$ and $Sent_2(R) = Sent_2(B) = -$
- Individual 3: $Sent_3(R) = Sent_3(B) = 0$ and $Sent_3(G) = -$

Using approval voting as a definition of the collective sentiment we obtain G as the most preferred entity, with two positive opinions received. With the same method we can easily construct a collective ranking of the entities, obtaining R and B tied in the second position with just one positive opinion received. ■

5.2.3 The voting theory approach

While sentiment analysis focuses only on polarities, the other extreme of the spectrum is the approach of voting theory, that is restricted to comparative preference information only. For the purpose of this work we represent individual preferences by using preorders, i.e., reflexive and transitive binary relations. This choice is motivated by two important characteristics of preferences extracted from textual expressions:

- *Interpersonal incomparability*: Since individuals have very different styles of writing or attitudes towards judging the entities under consideration, we believe that scores or any other form of graded polarity cannot be compared across individuals. Therefore we argue in favor of an ordinal representation of both regular and comparative opinions.
- *Incompleteness*: Since preferences and sentiments are observed from individual expressions we cannot assume this information to be complete.

Formally, we can define the voting theory approach as follows:

Definition 5.2.3 Given the individual data $(\sigma_i, \leqslant_i^P, \leqslant_i^N)$ extracted from individual expressions $\mathscr{T}_i$, the *pure preference data* associated with it is a preordered set $(\mathscr{D}_i, \leqslant_i^{\mathscr{D}})$, where $\mathscr{D}_i = D_i \cup P_i \cup N_i$, defined as:

$$x \leqslant_i^{\mathscr{D}} y \Leftrightarrow \begin{cases} x \leqslant_i^P y \text{ and } x, y \in P_i & \text{or} \\ x \leqslant_i^N y \text{ and } x, y \in N_i & \text{or} \\ x \in N_i \text{ and } y \in P_i & \text{or} \\ \sigma_i(x) \leqslant \sigma_i(y) \text{ and } x, y \in D_i \end{cases}$$

Pure preference data is thus the union of the comparative opinions extracted from the ordinal relation entailed by σ_i, with the addition of all the binary comparisons between elements from P_i and elements from N_i.

■ **Example 5.4** The pure preference data associated with Example 5.2 is the following:

- Individual 1: $B \sim_1 G <_1 R$

- Individual 2: $B <_2 R <_2 G$
- Individual 3: $G <_3 B \sim R$

Where $G < R$ stands for $G \leqslant R$ and $R \nleqslant G$, and $G \sim R$ stands for $G \leqslant R$ and $R \leqslant G$. Using a straightforward adaptation of the Borda rule to preorders, we obtain $B < G < R$ as the collective ranking: R receives 4 points, one for each alternative that is strictly ranked below by one of the individuals, G receives 2 points, and B only 1 point. ■

5.2.4 Combining Sentiment with Preference

In this section we propose a novel structure that combines features from both the sentiment analysis approach and the voting theory approach presented in the previous two sections. On the one hand we take binary comparisons as central to our analysis, using preorders to represent comparative preferential information. On the other hand, we complement this representation with a classification of the alternatives into three disjoint sets representing the positive, negative and neutral polarity:

Definition 5.2.4 An *SP-structure (for Sentiment-Preference structure)* over a set of candidates $\mathscr{X}$ is a tuple $S = (\mathscr{P}, \mathscr{N}, \mathscr{Z})$, where $\mathscr{P}$, $\mathscr{N}$ and $\mathscr{Z}$ are disjoint subsets of $\mathscr{X}$, and both $\mathscr{P}$ and $\mathscr{N}$ are ordered respectively by preorders $\leqslant^{\mathscr{P}}$ and $\leqslant^{\mathscr{N}}$.

An SP-structure indicates the subsets of positive ($\mathscr{P}$), negative ($\mathscr{N}$) and neutral ($\mathscr{Z}$) candidates among the set of entities $\mathscr{X}$, and specifies a set of binary comparisons between positive or negative candidates. The remaining elements of $\mathscr{X} \setminus (\mathscr{P} \cup \mathscr{N} \cup \mathscr{Z})$ are those alternatives for which no information has been collected.

We obtain SP-structures from individual data as follows:

Definition 5.2.5 Let $(\sigma_i, \leqslant_i^P, \leqslant_i^N)$ be the individual data extracted from individual expressions $\mathscr{T}_i$. The SP-structure associated with it is the tuple $(\mathscr{P}_i, \mathscr{N}_i, \mathscr{Z}_i)$:

- $\mathscr{P}_i = P_i \cup D_i^+$ where $D_i^+ = \{x \in D_i \mid \sigma_i(x) > 0\}$
- $\mathscr{N}_i = N_i \cup D_i^-$ where $D_i^- = \{x \in D_i \mid \sigma_i(x) < 0\}$
- $\mathscr{Z}_i = \{x \in D_i \mid \sigma_i(x) = 0\}$

with preorder relations defined as follows:

$$x \leqslant_i^{\mathscr{P}} y \quad \Leftrightarrow \quad \begin{cases} x \leqslant_i^P y \text{ and } x, y \in P_i & \text{or} \\ \sigma_i(x) \leqslant \sigma_i(y) \text{ and } x, y \in D_i^+ \end{cases}$$

$$x \leqslant_i^{\mathscr{N}} y \Leftrightarrow \begin{cases} x \leqslant_i^N y \text{ and } x, y \in N_i & \text{or} \\ \sigma_i(x) \leqslant \sigma_i(y) \text{ and } x, y \in D_i^- \end{cases}$$

■ **Example 5.5** The SP-structures associated with Example 5.2 are the following:

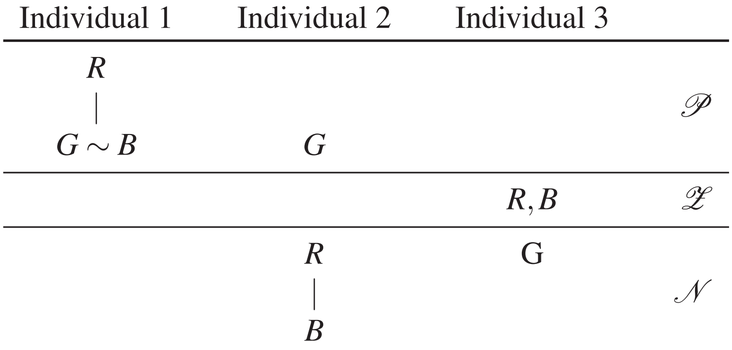

Figure 5.1: SP-structures associated with Example 5.2.

- Individual 1: $\mathscr{P}_1 = \{R, G, B\}$ with $G \leqslant_1^{\mathscr{P}} R$ and $G \sim_1^{\mathscr{P}} B$, $\mathscr{N}_1 = \mathscr{Z}_1 = \emptyset$
- Individual 2: $\mathscr{P}_2 = \{G\}$, $\mathscr{N}_2 = \{R, B\}$ with $B \leqslant_2^{\mathscr{N}} R$, $\mathscr{Z}_2 = \emptyset$
- Individual 3: $\mathscr{P}_3 = \emptyset$, $\mathscr{N}_3 = \{G\}$ and $\mathscr{Z}_3 = \{R, B\}$

SP-structures can be easily visualized, and in Figure 5.1 we draw the three structures described above. Alternatives that are in higher positions in the table are preferred to those that are in lower positions, and the three sets $\mathscr{P}$, $\mathscr{Z}$ and $\mathscr{N}$ are separated by horizontal lines. ■

In conclusion, an SP-structure compactly represents both sentiment information in the form of three polarity sets, as well as comparative opinions in the two preorders over the positive and negative sets. SP-structures are based on a purely ordinal view of preferences, hence assuming a very low degree of interpersonal comparability among individuals' preferences. This assumption could be relaxed by, for instance, normalizing the scores extracted from the individual data, or directly using them in the construction of the collective sentiment. While these approaches may fit some particular applications, they require additional assumptions to be able to merge comparative opinions, which are of an ordinal nature, with the possibly normalized scores representing regular opinions.

5.3 Borda counts for aggregating SP-structures

In order to aggregate SP-structures, and therefore put forward our definition of collective sentiment, in this section we define a class of aggregation procedures based on the classical Borda count (see Section 2.2). We begin by introducing a list of properties which are desirable in the context of sentiment analysis, and then put forward our definition of aggregation method. We show that this method satisfies all the desirable properties we introduced and that it generalizes both the existing definition used by sentiment analysis and the classical Borda rule in preference aggregation. We conclude the section with a study of the algorithmic aspects of our proposed Borda count.

5.3.1 Desired Axiomatic Properties

In this section we adapt classical axiomatic properties from the literature in social choice theory to the case of SP-structures, providing a suitable interpretation in the domain of sentiment analysis. We build on the axiomatization of the Borda rule proposed by Young [103], which we complement with axioms specific to our domain of application.

We first need to introduce some useful notation. Let us call a collection of SP-structures $(P_1,\dots,P_n)$ a *profile*, which we denote by P. If S_1 and S_2 are profiles of SP-structures, let S_1+S_2 be the profile obtained by putting together the two original profiles (renaming voters if necessary), i.e. if $S_1=(P_1,\dots,P_n)$ and $S_2=(Q_1,\dots,Q_m)$, then $S_1+S_2=(P_1,\dots,P_n,Q_1,\dots,Q_m)$. A profile is called *symmetric* if the set of individuals can be partitioned in pairs of individuals $\{i,i'\}$ with completely opposite SP-structures, i.e., if $\mathscr{P}_i=\mathscr{N}_{i'}$, $\mathscr{N}_i=\mathscr{P}_{i'}$, $\leqslant_i^{\mathscr{P}}=\overleftarrow{\leqslant_{i'}^{\mathscr{N}}}$ and $\leqslant_i^{\mathscr{N}}=\overleftarrow{\leqslant_{i'}^{\mathscr{P}}}$, where $\leqslant_i^{\mathscr{P}}=\overleftarrow{\leqslant_{i'}^{\mathscr{N}}}$ means that the preorder over the set $\mathscr{P}$ for voter i is equal to the inverted preorder over the set $\mathscr{N}$ of voter i'. A symmetric profile necessarily contains an even number of SP-structures. Finally, given a single SP-structure $S=(\mathscr{P},\mathscr{N},\mathscr{X})$, we say that a *voter i ranks a above b* in S if one of the following four conditions holds: $b\leqslant_i^{\mathscr{P}}a$ and $a\not\leqslant_i^{\mathscr{P}}b$, or $b\leqslant_i^{\mathscr{N}}a$ and $a\not\leqslant_i^{\mathscr{N}}b$, or $a\in\mathscr{P}$ and $b\in\mathscr{X}\cup\mathscr{N}$, or $a\in\mathscr{X}$ and $b\in\mathscr{N}$.

Let F be a rule which associates a set of most preferred alternatives with a profile of SP-structures. We now list a number of desirable properties for such an aggregation rule F.

The first set of properties is an adaptation of classical axioms from social choice theory, regarding the *equality* of treatment of alternatives and individuals:

- *Neutrality: For any profile S and permutation of entities $\rho:\mathscr{X}\to\mathscr{X}$, we have that $F(S_\rho)=\rho(F(S))$, where S_ρ is profile S with alternatives in $\mathscr{X}$ renamed by ρ.*

- *Anonymity: For any profile S and permutation of individuals $\rho:\mathscr{I}\to\mathscr{I}$, we have that $F(S_{\rho(1)},\dots,S_{\rho(n)})=F(S_1,\dots,S_n)$.*

Neutrality requires that if we rename the items in $\mathscr{X}$, the result should be the renaming of the initial result. Anonymity instead formalizes the fact that the collective opinion should not depend on the name of the individuals. The following two properties provide requirements on how to treat *consensus and total disagreement* in the individual preferences:

- *Weak-Pareto: If S is a profile in which all individuals rank a above b, then $b\notin F(S)$.*

- *Cancellation. If a profile S is symmetric then all entities are in the winning set, i.e., $F(S)=\mathscr{X}$.*

The weak-Pareto property is a fundamental property when aggregating individuals' preferences: agreement among all individuals should be reflected in the collective opinion.

Cancellation requires instead that if the disagreement is so extreme, as in a symmetric profile where individuals come in pairs whose preferences cancel each other out, then all items should be declared as the most preferred ones in the collective opinion.

The following two properties formalize the requirement that more information collected over an individual's preferences should lead to a result that is more preferred by that individual:

- *Voters participation: For all profiles $S = (S_1, \dots, S_n)$ and SP-structure S_{n+1}, any candidate in $F(S + S_{n+1}) \setminus F(S)$ is ranked in higher or same position than any candidate in $F(S)$ in the preferences of voter $n+1$.*

- *Rank participation: For all profiles $S = (S_1, \dots, S_n)$ and SP-structure $S' \subset S_n$, any candidate in $F(S) \setminus F(S_{-n} + S')$ is ranked in higher or same position than any candidate in $F(S_{-n} + S')$ in the preferences of voter n.*[1]

Where $S' \subset S_n$ means we are considering a subset of entities and their relative positions and relations do not change in the SP-structure. The property would express that each voter has incentive in expressing more information about as many entities as possible.

In the classical voting theory context, participation means that voters have an incentive to participate [78]. In a sentiment analysis context individuals have already expressed their opinion, so we do not need to favor their participation. However, the first property tells us that considering one more individual in the computation of the collective sentiment should result in a candidate that is higher in her ranking. In a similar way, the second property requires that more information on an individual's opinion lead to results that the individual ranks higher.

Finally, the following property formalizes the possibility of using techniques such as *map-reduce* [28] in particular profiles, for a more efficient computation of the set of most preferred alternatives:

- *Consistency: For all profiles S_1 and S_2, if $F(S_1) \cap F(S_2) \neq \emptyset$ then $F(S_1 + S_2) = F(S_1) \cap F(S_2)$.*

Consistency can be an important property in an application domain where one needs to deal with big quantities of data, such as sentiment analysis. It tells us that if we manage to partition the (possibly very large set of) individual opinions into smaller sets which have some best candidate in common, perhaps by means of a proper heuristic, then we can work on the elements of the partition independently. Thus *divide and conquer* approaches are possible, which parallelize and possibly speed up the computation.

All properties presented above are adapted from the literature on social choice theory

[1] Inclusions of SP-structures is defined as inclusions of preorders, and S_{-n} represents profile S without SP-structure S_n.

[6]. Not all combinations of axiomatic properties are feasible: for instance, the well-known Arrow's Theorem showed that it is not possible to aggregate linear orders using a rule that satisfies three simple desirable properties (namely, a weaker version of anonymity, an additional property called independence of irrelevant alternatives, and weak-Pareto) [5]. This is not the case for the list of axioms presented above, as we will show in the following sections.

5.3.2 The $B^*_{\underline{\alpha}}$ Rule

In this section we propose a parameterized class of aggregation procedures for profiles of SP-structures that builds on the classical Borda count, taking into account the incompleteness of the ordering and the additional information given by the sentiment polarity expressed by the individuals.

In the following definition we use $\underline{\alpha} = (\alpha_1, \alpha_2, \alpha_3, \alpha_4)$ with $\alpha_1, \alpha_2, \alpha_3, \alpha_4 \in \mathbb{R}^+$. If c in $\mathscr{P}$, $\mathrm{down}^{\mathscr{P}}(c)$ is the set of elements of $\mathscr{P}$ that are less preferred than c, $\mathrm{up}^{\mathscr{P}}(c)$ is the set of elements of $\mathscr{P}$ that are more preferred than c, and $\mathrm{inc}^{\mathscr{P}}(c)$ is defined as the the set of elements that are incomparable to c in $\mathscr{P}$ (in $\mathscr{N}$, respectively, for $\mathrm{down}^{\mathscr{N}}$, $\mathrm{up}^{\mathscr{N}}$ and $\mathrm{inc}^{\mathscr{N}}$). We will omit the reference to S when it is clear from the context.

Definition 5.3.1 Given an SP-structure $S = (\mathscr{P}, \mathscr{N}, \mathscr{Z})$ over $\mathscr{X}$, the $s^*_{\underline{\alpha}}$*-score* of an entity $c \in \mathscr{X}$ in S is defined as follows:

$$s^*_{\underline{\alpha}}(c,S) = \begin{cases} \alpha_1|\mathrm{down}^{\mathscr{P}}(c)| + \alpha_2|\mathrm{inc}^{\mathscr{P}}(c)| + \alpha_3|\mathscr{Z}| + \alpha_4 & \text{if } c \in \mathscr{P} \\ -\alpha_1|\mathrm{up}^{\mathscr{N}}(c)| - \alpha_2|\mathrm{inc}^{\mathscr{N}}(c)| - \alpha_3|\mathscr{Z}| - \alpha_4 & \text{if } c \in \mathscr{N} \\ 0 & \text{if } c \notin \mathscr{P} \cup \mathscr{N} \end{cases}$$

The $s^*_{\underline{\alpha}}$-score is defined as a parametrized class of scoring functions over SP-structures. It combines the approach from sentiment analysis, giving α_4 points to each alternative in the positive set and $-\alpha_4$ to all those in the negative set, with a generalization of the classical Borda rule, giving α_1 points to an alternative for all those that are ranked below, α_2 points for those ranked incomparable, and α_3 points for those alternatives that are in the neutral set (negative points if the alternative is in the negative set).

Note that no point is given to entities for which an individual has a neutral sentiment or for which she does not have any opinion. We are hence assuming that all score variables are initialized to 0, while an equivalent formulation could leave unspecified the score of alternatives for which no opinion has been extracted. The main difference between alternatives in $\mathscr{Z}$ and alternatives in $\mathscr{X} \setminus (P \cup \mathscr{N} \cup \mathscr{Z})$ is that the former do contribute to the score of alternatives in the positive or in the negative set via the parameter α_3, while the latter are not taken into consideration into the construction of the score of an alternative.

The use of a score may at first seems counter-intuitive given our discussion in Section 5.2 on the interpersonal incomparability of preferential information. However, what is being represented in the $s^*_{\underline{\alpha}}$-score is purely ordinal information about the *number* of alternatives being more or less preferred to others, and should not be confused with the *intensity* of preference that could have been expressed in the individual data via the scoring function σ_i.

To exemplify the flexibility of our setting, we can consider several assumptions on the $\underline{\alpha}$ vector that the modeler can choose, depending on the application at hand. For instance, assuming $\alpha_1 > \alpha_2$ and $\alpha_3 > \alpha_2$ will make sure that more points are given to alternatives that are strictly preferred to others than to those that are incomparable. Another possibility is to assume that the score difference between two successive elements in the positive or negative part should be less than the score difference between the least positive and the best negative elements, for instance when $2\alpha_4 > \alpha_1$. If these two numbers were equal, then the $s^*_{\underline{\alpha}}$-score would be equivalent to the classical Borda score when the set $\mathscr{Z}$ is empty, disregarding the sentiment information.

Definition 5.3.2 The score of an entity $c \in \mathscr{X}$ in the profile of SP-structures $S = \{S_i = (\mathscr{P}_i, \mathscr{N}_i, \mathscr{Z}_i) \mid i \in \mathscr{I}\}$ is defined as follows:

$$S^*_{\underline{\alpha}}(c,S) = \sum_{i \in \mathscr{I}} s^*_{\underline{\alpha}}(c,S_i)$$

where $s^*_{\underline{\alpha}}(c,S_i)$ is $s^*_{\underline{\alpha}}$-score of alternative c in the SP-structure S_i. The winners of the $B^*_{\underline{\alpha}}$ *rule* are the candidates with maximal total score:

$$B^*_{\underline{\alpha}}(S) = \operatorname*{argmax}_{c \in \mathscr{X}} S^*_{\underline{\alpha}}(c,S)$$

■ **Example 5.6** Let the parameters in $\underline{\alpha}$ be $\alpha_1 = \alpha_4 = 2$ and $\alpha_2 = \alpha_3 = 1$, and therefore let the corresponding score be as follows

$$s^*_{(2,1,1,2)}(c,S) = \begin{cases} 2 \times |\text{down}^{\mathscr{P}}(c)| + |\text{inc}^{\mathscr{P}}(c)| + |\mathscr{Z}| + 2 & \text{if } c \in \mathscr{P}_i \\ -2 \times |\text{up}^{\mathscr{N}}(c)| - |\text{inc}^{\mathscr{N}}(c)| - |\mathscr{Z}| - 2 & \text{if } c \in \mathscr{N}_i \\ 0 & \text{if } c \notin \mathscr{P}_i \cup \mathscr{N}_i \end{cases}$$

The winner of $B^*_{(2,1,1,2)}$ on the profile of SP-structures associated with Example 5.2 is R. Indeed, the score $s^*_{(2,1,1,2)}(R) = 4$ since there are 2 elements ranked below R in the positive part by the first individual ($4+2$ points), and R is ranked in the negative side by individual 2 (-2 point). G follows with a score of 1 since there is one element incomparable in the positive part by the first individual (1 + 2 points), G is ranked in the positive side by individual 2 (+2 point) and G is ranked in the negative side by individual 3 with 2 elements in the neutral set (-2-2 points). B is the worst preferred alternative with a score of -1, obtaining 3 points by individual one, -4 points by the second individual and 0 points by the third individual. ▪

5.3.3 Axiomatic analysis

We begin by showing that our Borda count generalizes the existing approaches used by sentiment analysis and preference aggregation. Let us first introduce some notation. Call a profile *purely preferential* if, for all $i \in \mathscr{I}$, the set $\mathscr{P}_i$ is equal to $\mathscr{X}$ and is linearly ordered, i.e., $\leqslant_i^{\mathscr{P}}$ is anti-symmetric, transitive and complete. Call a profile *purely non-negative sentimental* if for all $i \in \mathscr{I}$ the two sets $\mathscr{P}_i$ and $\mathscr{Z}_i$ form a partition of $\mathscr{X}$, and the candidates in $\mathscr{P}_i$ are all incomparable, i.e., the relation $\leqslant_i^{\mathscr{P}}$ is empty, and the set $\mathscr{N}_i$ is also empty. We now show that $B^*_{\underline{\alpha}}$ coincides with the Borda rule on purely preferential profiles, and that it coincides with approval voting on purely non-negative sentimental ones.

Theorem 5.3.1 If a profile S is purely preferential, then for all $\underline{\alpha}$ we have that $B^*_{\underline{\alpha}}(S) = \text{Borda}(S)$. If a profile S is purely non-negative sentimental, then for all $\underline{\alpha}$ such that $\alpha_2 = \alpha_3$ we have that $B^*_{\underline{\alpha}}(S) = \text{Approval}(S)$.

Proof. Let S be a purely preferential profile, i.e., all individuals are expressing a linear order over entities in $\mathscr{X}$ which are all in $\mathscr{P}$. Let $S^B(c,B)$ be the classical Borda score, i.e., the number of candidates ranked below c. Since in a purely preferential profile there are no alternatives that are incomparable to each other, and the sets $\mathscr{N}$ and $\mathscr{Z}$ are empty, the score $S^*_{\underline{\alpha}}(c) = \alpha_1 S^B(c) + \alpha_4 n$, where n is the number of voters. Since n is constant then the two rules elect the same candidates, no matter the value of α_1 and α_4.

Let now S be a purely non-negative sentimental profile and let $S^A(c)$ be the approval score of an entity c, i.e., the number of individuals approving c. Since all alternatives in $\mathscr{P}_i$ are incomparable, every approved entity in each single SP-structure gets a score equal to $\alpha_2(|\mathscr{X}|-1)+\alpha_4$. To see this, it is sufficient to observe that alternatives in $\mathscr{Z}$ give α_3 points to alternatives in $\mathscr{P}$ and $\alpha_2 = \alpha_3$, and moreover in a purely non-negative sentimental profile the two sets $\mathscr{P}$ and $\mathscr{Z}$ form a partition of $\mathscr{X}$. Hence, we obtain that $S^*_{\underline{\alpha}}(c) = (\alpha_2(|\mathscr{X}|-1)+\alpha_4) \cdot S^A(c)$ and thus $B^*_{\underline{\alpha}}$ elects the same candidates as approval voting. ■

Theorem 5.3.1 formalizes the fact that our proposed Borda count is a generalization of both approaches at the extreme of the spectrum described in Section 5.2: a pure sentiment analysis approach, which uses approval voting, and a pure preference aggregation approach, as described by the Borda rule. The result of Theorem 5.3.1 can be generalized to profiles of partial orders to show that $B^*_{\underline{\alpha}}$ extends the partial Borda count defined by Cullinan, Hsiao and Polett [26] as well as the *bucket averaging method* of Fagin et al. [38].

We now show that our proposed Borda count for collective sentiment analysis satisfies all the axiomatic properties presented in Section 5.3.1.

Theorem 5.3.2 $B^*_{\underline{\alpha}}$ satisfies consistency, neutrality, anonymity, voters participation, rank participation, and cancellation for all $\underline{\alpha}$. If we assume that $\alpha_1 \geqslant \alpha_2$, then $B^*_{\underline{\alpha}}$ also satisfies weak-Pareto.

Proof. For the sake of clarity we omit the reference to $\underline{\alpha}$ where it is not necessary. To prove that $B^*_{\underline{\alpha}}$ satisfies *consistency* it is sufficient to observe that $S^*_{S_1+S_2}(c) = S^*_{S_1}(c) + S^*_{S_2}(c)$. Those entities with maximal score in both S_1 and S_2 are then the entities with maximal score in $S_1 + S_2$. *Neutrality* and *anonymity* are straightforward consequences of our definition of $S^*_{\underline{\alpha}}$.

A simple monotonicity argument can be used to prove both versions of *participation*. Consider first voters-participation. Let $w \in B^*_{\underline{\alpha}}(S)$, and let S_{n+1} be the additional SP-structure. We show that the winner of the joint profile w' is not worse in $n+1$'s ranking than w. Since we have only added information from $n+1$, w' must have received strictly more points than w to become the new winner, and this can only happen if agent $n+1$ prefers w' to w. The same argument can be straightforwardly adapted to the case of rank-participation.

Finally, to prove that $B^*_{\underline{\alpha}}$ satisfies *cancellation* we observe that in a symmetric profile all entities have score 0, since $S^*_{\underline{\alpha}}$ is symmetric with respect to $\mathscr{P}$ and $\mathscr{N}$.

For *weak-Pareto*, there are four cases for an individual to rank a above b, and we can show that in all cases $s^*_{\underline{\alpha}}(a) > s^*_{\underline{\alpha}}(b)$ and thus that b cannot be in the winning set. Recall that we assumed $\alpha_1 \geqslant \alpha_2$. Assume that $b \leqslant_i^{\mathscr{P}} a$ and $a \not\leqslant_i^{\mathscr{P}} b$. If a third alternative c is ranked below b then by transitivity c is also ranked below a, and hence a and b get the same points from c. If $c \in \text{inc}^{\mathscr{P}_i}(b)$ or $c \in \mathscr{Z}_i$ then this also gives the same points to a (or more, if c is ranked below a since $\alpha_1 \geqslant \alpha_2$). Finally, $\text{down}^{\mathscr{P}_i}(b) \subset \text{down}^{\mathscr{P}_i}(a)$ and thus a gets α_1 more points than b. The case in which a and b are both in $\mathscr{N}_i$ is treated symmetrically: just consider alternatives ranked above the two and the set $\text{up}^{\mathscr{N}_i}(a)$ rather than $\text{down}^{\mathscr{P}_i}(a)$. If we instead assume that $a \in \mathscr{P}_i$ and $b \in \mathscr{Z}_i \cup \mathscr{N}_i$, then it is easy to observe that $s^*_{\underline{\alpha}}(a) > 0$ while $s^*_{\underline{\alpha}}(b) \leqslant 0$ and thus that also in this case $s^*_{\underline{\alpha}}(a) > s^*_{\underline{\alpha}}(b)$. Finally, if $a \in \mathscr{Z}_i$ and $b \in \mathscr{N}_i$ then $s^*_{\underline{\alpha}}(a) = 0$ but $s^*_{\underline{\alpha}}(b) < 0$ since $b \in \mathscr{N}_i$ gets $-\alpha_4$ points and any other alternatives $c \in \mathscr{N}_i$ can only decrease the score of b. ■

We conclude this section by comparing $B^*_{\underline{\alpha}}$ with another rule, introduced in previous work by Brams and Sanver [16], that aims at combining approval voting with preference aggregation: *fallback voting*. Under this rule, each voter approves a subset (which could be empty) of candidates and ranks them in a linear order. The winner of fallback voting is obtained in an iterative way, by first checking whether there is a candidate that is top-ranked by a majority of voters. If such a candidate does not exist, then the first and the second ranked candidates in each voters' preference are considered, and once again it is checked if there is a candidate that is ranked first or second by a majority of the voters. The process goes on adding the third and subsequently ranked candidates until an alternative obtains a majority of approvals. If the process ends finding any candidate approved by a majority of voters then candidates with the highest number of approval are the winners. The structures used by fallback voting to combine approvals with comparative preferences can be seen as a special case of SP-structures, with no neutral nor negative sets and linearly ordered items. Fallback voting may result in a different outcome than $B^*_{\underline{\alpha}}$. A detailed study of the difference between these two rules is left as future work. While fallback voting constitutes an interesting voting rule when individuals have incentives to express their preferences, in applications such as sentiment analysis the individual data that is collected will rarely be complete (see Section 5.2). Hence the need for rules that are able to handle incomplete profiles, such as our proposed $B^*_{\underline{\alpha}}$.

5.3.4 Algorithmic properties of $B^*_{\underline{\alpha}}$

In this section we analyze the algorithmic aspects of our proposed Borda count for collective sentiment analysis. Given the envisioned application, it is important that the basic problem of computing the most preferred alternative in a given profile be tractable, i.e. that the computational complexity of winner determination be solvable in polynomial time. We also provide an exact bound on the minimum number of bits required to compute the outcome of $B^*_{\underline{\alpha}}$ (aka. its communication complexity) and we show that it can be computed with an incremental algorithm.

Let us first make some considerations about the size of representing an SP-structure. Recall that we start from a set of m alternatives or entities $\mathscr{X}$, and n individuals. We assume that the sets $\mathscr{N}$, $\mathscr{P}$ and $\mathscr{Z}$ are encoded with a vector of length m containing for each alternative in $\mathscr{X}$ a label of 2 bits for the set to which it belongs to. Since the set $\mathscr{N}$ and $\mathscr{P}$ are disjoint, the two preorders $\leqslant^{\mathscr{P}}$ and $\leqslant^{\mathscr{N}}$ can be represented as binary relations on an $m \times m$ matrix, indicating for every pair (a,b) whether $a \leqslant^{\mathscr{P}} b$ or $a \leqslant^{\mathscr{N}} b$. The size of a profile of SP-structures with n individuals is therefore $O(nm^2)$.

The problem of *winner determination* is the algorithmic task of deciding whether a designed alternative $a \in \mathscr{X}$ is in the winning set of a given profile of SP-structures S. This problem has been widely studied in voting theory [19], where its tractability is often considered a requirement for a rule to be considered of practical interest. We now show that the winner of our proposed Borda count can be computed in polynomial time:

Theorem 5.3.3 The winner of $B^*_{\underline{\alpha}}$ can be computed in time $O(nm^2)$, hence in time linear in the size of the input.

Proof. Given a single SP-structure S_i, the $s^*_{\underline{\alpha}}$-score of an alternative a can be computed in the following way. First check whether $a \in \mathscr{Z}_i$, $a \in \mathscr{P}_i$, or $a \in \mathscr{N}_i$, which can be done in constant time. If $a \in \mathscr{Z}_i$ then its score is 0. Otherwise we can compute its score by first counting how many alternatives are in $\mathscr{Z}_i$, which can be done in $O(m)$, and then counting how many alternatives are ranked below a, in case $a \in \mathscr{P}_i$, or how many alternatives are ranked above a, if $a \in \mathscr{N}_i$, and finally how many alternatives are incomparable to a. All these operations can be done in $O(m)$. We repeat this process for each alternative a and for each individual i, obtaining the upper bound $O(nm^2)$. ■

A further important algorithmic property of a voting procedure is its *communication complexity*, i.e. the minimal amount of bits that needs to be expressed by the individuals in order to compute the most preferred alternatives. Previous work [24] provided lower and upper bounds for many voting procedures including the Borda rule, showing that the communication complexity of computing the Borda winner is $\Theta(nm\log m)$, i.e., lower and upper bounds are both equal, up to multiplication by a constant, to the function $nm\log m$.Since our Borda count for collective sentiment analysis generalizes the classical Borda rule (see Theorem 5.3.1), the communication complexity is within a constant factor of the traditional setting:

Theorem 5.3.4 The communication complexity of $B^*_{\underline{\alpha}}$ is in $\Theta(nm\log m)$.

Proof. An upper bound is easy to obtain, since it is sufficient for each individual to specify their SP-structure to be able to compute the winner of $B^*_{\underline{\alpha}}$. Given our representation of SP-structures, a profile can be specified using a matrix $n \times m$, where each cell (i, j) contains a number $x \in \mathbb{Z}$. If $x > 0$ then voter i has a positive opinion for candidate j and x indicate how many candidates are ranked below her in the SP-structure, if $x < 0$ then voter i has a negative opinion for candidate j and x indicate how many candidates are ranked higher in the SP-structure, if $x = 0$ then voter i has a neutral opinion for candidate j. Moreover if there exist candidates that are incomparable in an SP-structure then they will have the same value. So the higher position that a candidate can have in any SP-structure is $m-1$ which can be represented using $\log(m-1)$ bits and we need $m \times n$ of these value, which proves that the upper bound for the communication complexity is $O(nm\log m)$. A lower bound can instead be obtained by adapting the same bound for the classical Borda rule provided in [24]. ■

We conclude the section by proposing a notion of *incremental complexity* that should capture the feasibility of computing the result of an aggregation procedure in an on-line fashion. This is a very important aspect when data is examined incrementally or when using methods such as map-reduce [28] to deal with large quantities of data. Recall the two participation axioms we introduced in Section 5.3.1: they imply that the result of an aggregation procedure should take into consideration the additional information collected from the individuals. A good aggregation procedure that can be used to define a notion of collective sentiment should not only take care of this additional information, but also be able to update the outcome in little time.

Let $\mathscr{X}$ be a set of alternatives. An *individual expression* P over $\mathscr{X}$ is a preference, a vote, an opinion, or an SP-structure defined on $\mathscr{X}$. Given a profile of individual expressions $(P_1,\ldots,P_n)$ – be it a profile of linear orders, of approval sets or of SP-structures – a *generalized voting rule* F is a function that outputs a set of most preferred alternatives $F(P_1,\ldots,P_n) \subseteq \mathscr{X}$. If we denote with $\mathscr{PR}$ the space of all possible profiles for all finite n, then $F : \mathscr{PR} \to 2^{\mathscr{X}}$. The Borda rule, approval voting, fallback voting and $B^*_{\underline{\alpha}}$ are all generalized voting rules. We give the following definition:

Definition 5.3.3 An generalized voting rule F is *incremental* if there exists a representation of profiles $r : \mathscr{PR} \to \{0,1\}^*$ and a function $\hat{F} : \{0,1\}^* \times \mathscr{PR}^1 \to 2^{\mathscr{X}}$, where $\mathscr{PR}^1$ is the set of all individual expressions, such that:

- $F(P_1,\ldots,P_{n+1}) = \hat{F}(r(P_1,\ldots,P_n),P_{n+1})$ for all profiles $(P_1,\ldots,P_{n+1}) \in \mathscr{PR}$ and all finite n;
- for every sequence of individual expressions $\{P_i \mid i \in \mathbb{N}\}$, the following holds:

$$\lim_{n\to+\infty} \frac{size[r(P_1,\ldots,P_n)]}{size[(P_1,\ldots,P_n)]} = 0$$

A generalized voting rule is incremental if its outcome, i.e. the set of most preferred candidates, can be computed by receiving the individuals expressions in a sequence, at each step computing the new outcome and storing a minimal amount of information that is needed to compute the outcome of the following step. Moreover, we require that the information stored at each step using function r be much smaller than the full representation of a profile as n grows.

Let us first show that both the Borda rule and approval voting are incremental. The Borda rule can be computed incrementally by storing the total Borda score of each alternative, and this can be done in space $O(m\log(nm))$ since $n \times (m-1)$ is the maximal Borda score that an alternative can obtain. When additional information is collected in the form of a linear order P_{n+1}, the total Borda scores can simply be updated with the additional scores computed. Since the size of a profile of linear orders is $O(nm\log m)$, the requirement on the size of the representation holds:

$$\lim_{n\to+\infty} \frac{size[r(P_1,\ldots,P_n)]}{size[(P_1,\ldots,P_n)]} = \lim_{n\to\infty} \frac{O(m\log(nm))}{O(nm\log m)} = \lim_{n\to\infty} \frac{\log n}{n} = 0$$

Approval voting receives as input a profile of sets of approved candidates, which has size $O(nm)$. An incremental procedure for its computation stores the number of approvals received by any candidate, and updates them when a new voter submits her ballots. Hence $size[r(P_1,\ldots,P_n)] = O(m\log n)$, showing that approval voting is also an incremental aggregation procedure.

Let us conclude by showing that $B^*_{\underline{\alpha}}$ is incremental for any $\underline{\alpha}$. In the same way as the Borda rule, the total score obtained by each alternative in a profile of SP-structures S can be stored in space $O(m\log(nm))$, since the maximal $S^*_{\underline{\alpha}}$-score is a linear function of $n\times m$. When a new SP-structure S is extracted, the total scores can be updated and the new outcome computed. Since a profile of SP-structures is represented in space $O(nm^2)$ we obtain the following:

$$\lim_{n\to+\infty} \frac{size[r(P_1,\ldots,P_n)]}{size[(P_1,\ldots,P_n)]} = \frac{m\log(nm)}{nm\log m} = 0$$

We conjecture that all anonymous voting rules are incremental by Definition 5.3.3. In fact, if a voting rule is anonymous then the information contained in a profile with n voters can be summarized by using an amount of space that grows sub-linearly with n. However, our definition measures an important characteristic of a generalized voting rule, one that is particularly useful in applications that deal with large numbers of individuals. The notion of incrementality we proposed resembles the *on-line time* discussed by Maudet et Al. [73], which however focuses mostly on space complexity requirements. A complete study of the notion of incrementality in generalized voting rules is beyond the scope of this work, but the above discussion may serve as a starting point for further work on this topic.

5.4 Empirical Analysis

This section reports on our experimental evaluation of the $B^*_{\underline{\alpha}}$ rule proposed in Section 5.3. The problem we face in this section is two-fold: First, in order to assess the relevance of preferential ordering information in determining the collectively preferred alternatives, we present experiments showing that there is a significant difference between the classical definition used by sentiment analysis and the result of the $B^*_{\underline{\alpha}}$ rule. Second, given the information sparsity which is characteristic of sentiment analysis domains, we evaluate the accuracy of the $B^*_{\underline{\alpha}}$ rule in situations of incompleteness, showing that its accuracy grows linearly with the amount of information available. In all our experiments we fix the parameters of the $B^*_{\underline{\alpha}}$ rule to $\underline{\alpha} = (2,1,1,2)$. We start this section with an analysis which motivate the empirical analysis.

5.4.1 Analytical analysis

Counting the number of profiles where sentiment analysis and preference aggregation return a different winner can be a difficult task. We try to analytically describe the case with only two candidates and so where $C = \{a,b\}$, where the number of voters n is odd and using a tiebreaking rule that prefers a to b. In this case, the possible preferences that a single voter can cast are (preferences are represented using the same syntax of the Example 5.1 where candidates on the left side of the $|$ have a positive opinion, while the ones on the right side have a negative opinion):

1. a is preferred to b and the voter has a positive opinion for both of them ($ab|$)
2. a is preferred to b and the voter has a positive opinion for a and a negative opinion for b ($a|b$)
3. a is preferred to b and the voter has a negative opinion for both of them ($|ab$)
4. b is preferred to a and the voter has a positive opinion for both of them ($ba|$)
5. b is preferred to a and the voter has a positive opinion for a and a negative opinion for a ($b|a$)
6. b is preferred to a and the voter has a negative opinion for both of them ($|ba$)

We can identify two different types of profiles that make the two aggregation procedures return a different winner:

- profile where l (with $l \geqslant (n+1)/2$) voters have preferences of type 4 and 6 and where $t \leqslant m$, where t is the number of voters with preferences of type 5 and m the number of voters with preferences of type 2
- profile where l (with $l \geqslant (n+1)/2$) voters have preferences of type 1 and 3 and where $t > m$, where t is the number of voters with preferences of type 5 and m the number of voters with preferences of type 2

In the first case preferences of type 4 and 6 represent the majority of the voters, this makes a the preferred candidate for the sentiment analysis, since the majority gives the same sentiment support to both candidates, while b is the winner for preference aggregation. Preferences of type 5 can change the outcome of sentiment analysis in favour of b but in this case their support is cancelled by preferences of type 2. The remaining voters' preferences cannot change the winner of one of the two aggregation procedure. In fact, if they have preferences of type 1 or 3 they increase the sentiment support without changing the preferences for b. This first case can be formulated as follow:

$$\sum_{l=\frac{n+1}{2}}^{n} \binom{n}{l} \sum_{t=0}^{n-l} \binom{l}{t} 2^{l-t} \sum_{m=t}^{n-l} \binom{n-l}{m} 2^{n-l-m} \tag{5.1}$$

In the second case the majority of the voters prefers a to b and have the same opinion about the two. So while a is the winner for the preference aggregation procedure, they have the same support for the sentiment analysis. This can return a different winner only if the sentiment support is greater and this can be achieved only if some other voter has preferences of type 5 and voters with this type of preference are more than the voters with preferences of type 2. Once again the remaining voters' preferences cannot change the winner of one of the two aggregation procedure and this case ca be formulated as follow:

$$\sum_{l=\frac{n+1}{2}}^{n-1} \binom{n}{l} \sum_{m=0}^{n-l-1} \binom{l}{m} 2^{l-m} \sum_{t=m+1}^{n-l} \binom{n-l}{t} 2^{n-l-t} \tag{5.2}$$

The total amount of profiles where the two aggregation procedures return a different winner is the sum of formulas 5.1 and 5.2. Where binomials compute the possible combinations. For instance the first binomial in both formulas compute all the possible ways of composing a majority in a profile with n voters.

The characterization is not an easy task and also for this reason we decide to make an empirical analysis which highlights this situations.

5.4.2 Sentiment Analysis and Borda Count

A crucial factor supporting our claim that more complex models of preferences and aggregation procedures should be used in the definition of collective sentiment is that the classical sentiment analysis method (equivalent to approval voting) and $B^*_{\underline{\alpha}}$ output different results over the same data. In fact, if they did not differ enough, it would mean that the ordering information (not considered by approval/sentiment analysis) is not relevant for determining the winner. Thus there would be no point in extracting ordering information from individuals.

Figure 5.2 reports on our experiments on the simplest case of 2 candidates. We enumerated all profiles of totally ordered SP-structures with n voters, with n from 2 to 90, where a totally ordered SP-structure is an ordering over the two candidates (that is, a over b or b over a), plus a threshold which associates to each candidate either a positive or a negative sentiment. Thus, there are 6 possible such SP-structures. We have computed the winning candidates according to sentiment analysis (i.e, approval voting) and according to $B^*_{\underline{\alpha}}$, which in the case of 2 candidates is equivalent to using the majority rule, and we have counted the percentage of profiles on which the two winners are different. Figure 5.2 shows that such percentage stabilizes at around 30%.

We have also varied the number of candidates from 2 to 100, keeping the number of voters fixed at 10, in which case however we did not enumerate all possibilities but we

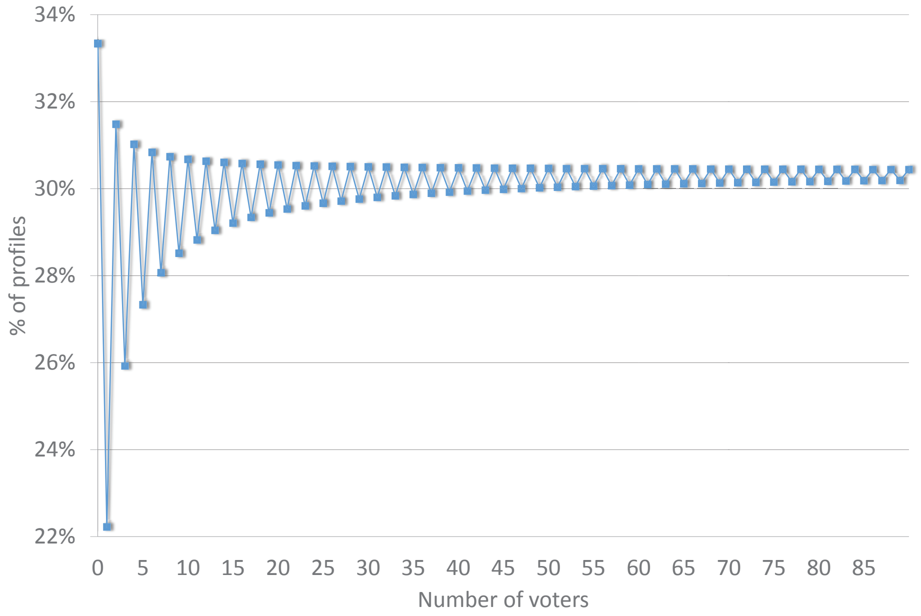

Figure 5.2: Percentage of profiles where sentiment analysis and $B^*_{\underline{\alpha}}$ differ (2 candidates).

generated 10.000 profiles of complete SP-structures with the impartial culture assumption, i.e., we sampled profiles with uniform distribution. Figure 5.3 shows that the percentage of cases where $B^*_{\underline{\alpha}}$ yields a different result than sentiment analysis grows with the number of candidates reaching more than 60%.

5.4.3 Incomplete Data

In practical applications individuals are likely to express their opinions over a small subset of the alternatives under considerations, as observed, e.g., in the studies conducted on the Netflix dataset [13]. It is therefore important to assess the behavior of our proposed Borda count on incomplete profiles.

To do this we generated profiles of complete SP-structures with 10 candidates and 100 voters, and we deleted a certain percentage of information to obtain an incomplete version of the profile. More precisely, we generated incomplete profiles in the following way: we first generated complete profiles and then we picked randomly a voter and a candidate, which is either positive or negative for that voter, and we changed the SP-structure of that voter to have no opinion on the selected candidate. With n voters and c candidates, nc corresponds to 100% of the information. Thus deleting $x\%$ of the information means performing the above described modification of the profile $(xnc)/100$ times. We then

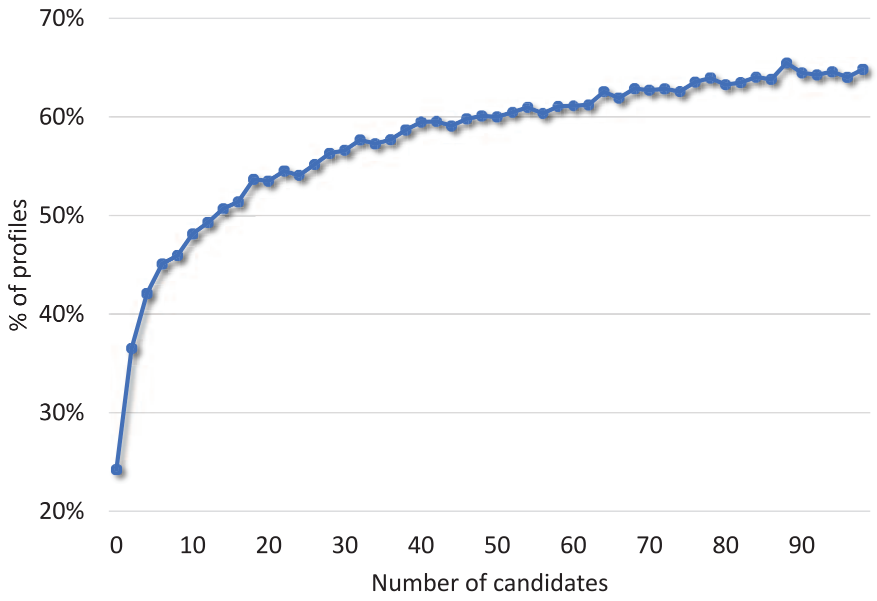

Figure 5.3: Percentage of profiles where sentiment analysis and $B^*_{\underline{\alpha}}$ differ (10 voters).

compared the winner (according to $B^*_{\underline{\alpha}}$) in the complete profile and in the incomplete one, by computing the absolute value of the difference between their $S^*_{\underline{\alpha}}$ scores, and we normalized it by dividing by the maximal error in the complete profile. Finally, we averaged over 10.000 profiles, obtaining the mean error introduced by the incompleteness of the profile. Figure 5.4 shows the trend in the error depending on the completeness of the profiles (mean error and variance). We also show the error of the random procedure, which outputs a candidate with uniform probability.

It is easy to see that Borda* always behaves better than the random procedure in identifying the winner in the complete profile, and moreover that its shape shows that accuracy quickly grows when the completeness of the profile increases.

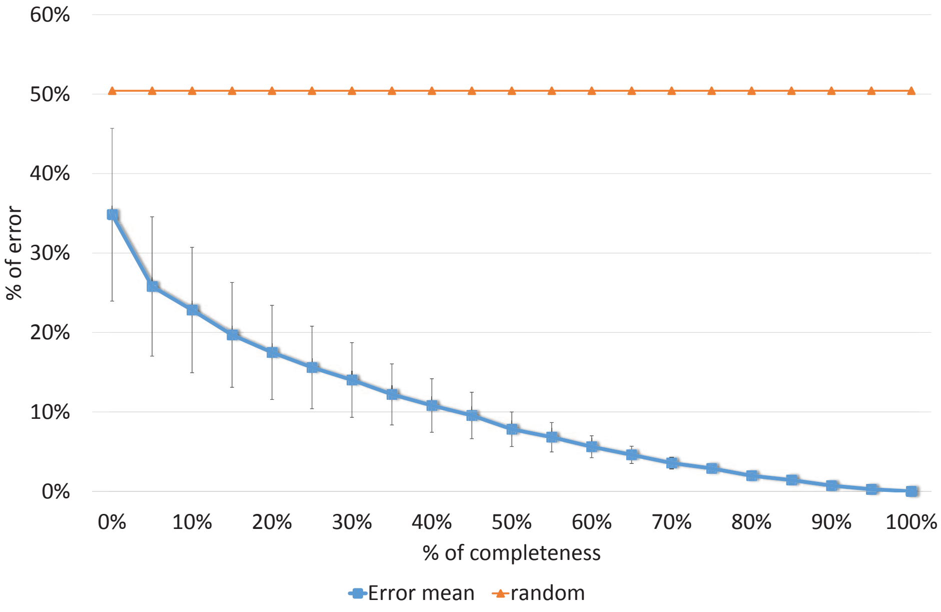

Figure 5.4: Mean error of $B^*_{\underline{\alpha}}$ on incomplete profiles.

6. Conclusions

The thesis tackles three different problems of the computational social choice area. The first two are more related one each other since both of them are related with the notion of acting strategically in a voting system.

In Chapter 3 we study and report the results about the computational complexity of replacement control. Using this kind of control action the chair tries to influence an election by replacing candidates or votes, either constructively or destructively, which means to favour a specific candidate or to ensure that a particular candidate does not win the election. Table 6.1 shows most of the complexity results for the various voting rules and types of replacement control analyzed in this chapter.

Control	Plurality	Veto	Borda	Approval	k-approval
CCRV	V	V	R	R	R $(2 < k < m-2)$
DCRV	V	V	V	V	V
CCRC	R	R	R	V	R
DCRC	R	R	V	V	R

Table 6.1: Summary of results (V: vulnerable, R: resistant).

Since the theoretical study is focused on worst-case results we also performed an experimental analysis, using real-world data sets to verify whether k-approval and Borda are really difficult to control (DCRC) in practice. While this study confirmed that Borda is easy to control (as expected), it also suggests that k-approval can be easy to control in practice despite our theoretical analysis classifying it as resistant. These results suggest that the study of computational complexity in the worst case is not enough to ensure a significant protection to the system and empirical evidence can be beneficial, for instance it can reveal how the control problem behaves when preferences are not uniform distributed.

Chapter 4 studied the effects of iterative strategic reasoning on classical voting rules by allowing individuals to manipulate the outcome of the election using a restricted set of manipulation moves. We provided two new definitions of manipulation moves, *second-chance* and *best-upgrade*, and showed that they lead to convergence for all voting rules considered (cf. Theorem 4.3.1 and 4.3.2) except for STV. We evaluated both theoretically and experimentally the performance of our restricted manipulation moves, with respect to the Condorcet efficiency of the iterative version of a voting rule as well as the Borda score of the winner in the initial truthful profile. We performed two simulations based on prototypical examples of iterated manipulation: the first simulation with the number of candidates smaller but comparable to the number of voters to model scheduling with Doodle, and the second with the number of voters much higher than that of candidates to model iterated polls before a political election. With the exception of the Borda rule, we showed that restricted manipulation in iterative voting yields a positive increase in both the Condorcet efficiency and Borda score and that the best performance is obtained when the number of candidates is higher than the number of individuals. We also showed that some axiomatic properties, such as unanimity and Condorcet consistency, are preserved in the iterative process.

A first direction for future work is the analysis of different versions of manipulation moves, and a comparison with ours and existing definitions. A starting point can be the work of Obraztsova and Elkind [81], in which polynomial manipulation moves based on distance are proposed. In this respect, parameters other than the Condorcet efficiency and the Borda score may be used to evaluate the effect and the costs of iteration. A second direction is the study of the computational complexity of manipulating iterated voting rules, as presented at the end of Section 4.5. A third direction is the application of the framework we defined in this chapter to account for preferences expressed as partial orders, and preferences expressed in combinatorial domains [22]. This generalization will allow us, among other things, to exploit preference data extracted from real-world elections and collective decision-making processes and to assess with more accuracy the effects of iterated restricted manipulation on less correlated distributions of preferences.

In Chapter 5 we proposed a definition of collective sentiment over multiple items inspired from existing research in voting theory and preference aggregation. By representing individuals' sentiment and preferences in a single structure we were able to encompass classical approaches, putting forward a generalization of the classical Borda count which has very good theoretical properties and behaves well in incomplete domains such as those distinctive of sentiment analysis and opinion mining.

Related work in in this research area have traditionally focused on aggregating preferences and opinions submitted by individuals in a uniform format, in the case of sentiments and comparative preferences extracted from individual textual expressions we have to deal with richer structures that are however incomplete. preferences and with the problem of combining preference and sentiments in an accurate definition of collective sentiment.

Our work opens several directions for future work. We list a number of challenges that arise from the use of techniques from preference aggregation and voting theory for collective sentiment analysis.

More refined models of opinions.

As already noted in Section 5.1, our analysis of preference and opinion extraction disregarded two important parameters:

- *Time*. Individual opinions are expressed at a given point in time and are also subject to change or updates. Hence, temporal information plays an important role in defining a coherent individual view. We believe that the literature on knowledge representation [98], in particular belief revision and merging, provides useful tools for the analysis and summarization of conflicting information that can be applied to the modeling of this problem.
- *Features*. Entities or items are usually described by means of features, i.e., they may be elements of a product space. Techniques from natural language processing can be used to extract the relevant features and thus build the set of entities. However, in this setting preferences and opinions may compare features rather than entities, requiring a more elaborate framework for its extraction and representation. Moreover, the combinatorial explosion resulting from a large set of features may give rise to computational problems that require an adequate compact representation framework for preferences. The literature on social choice in combinatorial domains [62] and in particular on judgment aggregation [33] is highly relevant to this problem.

Validation of aggregation rules.

Since the variety of preference aggregation methods that can be defined is very large, of which a prime example is the $B^*_{\underline{\alpha}}$ rule depending on the values given to its parameters $\underline{\alpha}$, a natural question is how to make a choice among them. Two options are possible, depending on the use of sentiment analysis techniques as a predictor for real-world events or as a tool for decision-making. First, if methods of collective sentiment analysis are used over time, tested for several settings and items, and employed in the context of predicting the result of real-world processes (such as elections or the evolution of a market, see, e.g., [4]), then *machine learning* techniques can be used to learn the best aggregation method, that is, the one that has proven to be the most accurate. Work on voting rules seen as *maximum likelihood estimators* can also be useful in this respect [24]. Alternatively, as in classical voting theory and as performed in this chapter, axiomatic properties as well as results about the computational complexity of aggregation rules could guide the choice of some aggregation methods over others.

Strategic behavior in sentiment analysis.

The individuals composing a society, as well as the agents in a multiagent system, are very often connected by interpersonal ties, e.g., when individuals are organised in a network. In this case, individual preferences and opinions are not only the result of personal reflection but may also take into consideration the position taken by influential individuals or simply by agents that are close to them in the network. The field of *social network analysis* [57, 31] is a burgeoning research area which has the potential of generating highly interesting results once combined with techniques of preference and sentiment analysis. Sentiment analysis techniques are moreover not immune to strategic manipulation. A rising phenomenon is the creation of web services proposing the opening of thousands of fake Twitter accounts to be used as followers of the manipulator's account, or the publishing of big volumes of positive posts related to the manipulator's products. This represents a prime example of strategic behavior in collective choice problems, and the whole body of literature published on this topic may be put to test with real world data once the two fields of sentiment analysis and preference aggregation have been put together to their full potential. While the problem of manipulation for a single agent is computationally easy for the classical case of the Borda rule, we conjecture that for $B^*_{\underline{\alpha}}$ this is not the case, given the higher amount of possibilities that an agent has to manipulate the election. However, single-agent manipulation is unlikely to occur in sentiment analysis applications, given the high number of individuals concerned and the absence of a well-defined elicitation protocol. Instead, an interesting direction for future work is the study of coalitional manipulation, which was recently shown intractable even for the classical Borda rule [27, 15].

Big data and collective sentiment analysis.

When the aggregation operation is relatively simple (e.g., the majority rule), it is possible to use straightforward techniques such as *Hadoop MapReduce* [28] to perform computations in parallel. The mapping phase can be used to run sentiment classifiers on text corpora in parallel; the resulting data objects can be combined/reduced in parallel. However, with more complex structures (e.g., conditional preference networks when the set of entities is described by means of features), the combination procedure may be more combinatorial in nature and may require non-trivial parallel processing. In this context, modern scale-out programming languages such as X10 can be particularly valuable [21], making it easy to write code that runs over thousands of cores and deals with hundreds of gigabytes of main memory data. Of particular interest is developing incremental parallel algorithms that can update collective sentiments as new utterances stream in and need to be processed.

Bibliography

[1] David J. Abraham, Avrim Blum, and Tuomas Sandholm. Clearing algorithms for barter exchange markets: enabling nationwide kidney exchanges. In Jeffrey K. MacKie-Mason, David C. Parkes, and Paul Resnick, editors, *ACM Conference on Electronic Commerce*, pages 295–304. ACM, 2007.

[2] Stéphane Airiau and Ulle Endriss. Iterated majority voting. In *Proceedings of the 1st International Conference on Algorithmic Decision Theory (ADT-2009)*, 2009.

[3] Krzysztof R. Apt and Sunil Simon. A classification of weakly acyclic games. In Maria Serna, editor, *SAGT*, volume 7615 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2012.

[4] Marta Arias, Argimiro Arratia, and Ramon Xuriguera. Forecasting with twitter data. *ACM TIST*, 5(1):8, 2013.

[5] Kenneth J. Arrow. *Social Choice and Individual Values*. John Wiley & Sons Inc., New York London Sydney, 1963.

[6] Kenneth J. Arrow, Amartya K. Sen, and Kotaro Suzumura, editors. *Handbook of social choice and welfare*. Elsevier, Amsterdam [u.a.], 1. ed. edition, 2002.

[7] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of LREC-2010*, 2010.

[8] Ramnath Balasubramanyan, Bryan R. Routledge, and Noah A. Smith. From tweets to polls : Linking text sentiment to public opinion time series. In *Proceedings of the International AAAI Conference on Weblogs and Social Media*, 2010.

[9] John J. Bartholdi and James B. Orlin. Single transferable vote resists strategic voting. *Social Choice and Welfare*, 8:341–354, 1991.

[10] John J. Bartholdi, Craig A. Tovey, and Michael A. Trick. How hard is it to control an election. *Mathematical and Computer Modeling*, pages 27–40, 1992.

[11] John J. Bartholdi, Craig A. Tovey III, and Michael A. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241, 1989.

[12] Dorothea Baumeister, Gábor Erdélyi, Olivia Johanna Erdélyi, and Jörg Rothe. Computational aspects of manipulation and control in judgment aggregation. In Patrice Perny, Marc Pirlot, and Alexis Tsoukiàs, editors, *ADT*, volume 8176 of *Lecture Notes in Computer Science*, pages 71–85. Springer, 2013.

[13] James Bennett and Stan Lanning. The Netflix prize. In *Proceedings of the KDD Cup and workshop*, 2007.

[14] Sven Berg. Paradox of voting under an urn model: The effect of homogeneity. *Public Choice*, 47(2):377–387, 1985.

[15] Nadja Betzler, Rolf Niedermeier, and Gerhard J. Woeginger. Unweighted coalitional manipulation under the borda rule is np-hard. In *Proceedings of IJCAI-2011*, 2011.

[16] Steven Brams and M. Remzi Sanver. Voting systems that combine approval and preference. In StevenJ. Brams, WilliamV. Gehrlein, and FredS. Roberts, editors, *The Mathematics of Preference, Choice and Order*. Springer-Verlag Berlin Heidelberg, 2009.

[17] Steven J. Brams and Peter C. Fishburn. Voting procedures. In Kenneth Arrow, Amartya Sen, and Kotaro Suzumura, editors, *Handbook of Social Choice and Welfare*. Elsevier, 2002.

[18] Steven J. Brams and Peter C. Fishburn. *Approval voting*. Springer, 2nd edition, 2007.

[19] Felix Brandt, Vincent Conitzer, and Ulle Endriss. Computational social choice. In G. Weiss, editor, *Multiagent Systems*, pages 213–283. MIT Press, 2013.

[20] Andrea Ceron, Luigi Curini, Stefano M. Iacus, and Giuseppe Porro. Every tweet counts? How sentiment analysis of social media can improve our knowledge of citizens' political preferences with an application to Italy and France. *New Media & Society*, 2013.

[21] Philippe Charles, Christian Grothoff, Vijay Saraswat, Christopher Donawa, Allan Kielstra, Kemal Ebcioglu, Christoph von Praun, and Vivek Sarkar. X10: an object-oriented approach to non-uniform cluster computing. In *Proceedings OOPSLA-2005*, 2005.

[22] Yann Chevaleyre, Ulle Endriss, Jérôme Lang, and Nicolas Maudet. Preference handling in combinatorial domains: From ai to social choice. *AI Magazine, Special Issue on Preferences*, 29(4):37–46, 2008.

[23] Vincent Conitzer and Tuomas Sandholm. Complexity of manipulating elections with few candidates. In Rina Dechter and Richard S. Sutton, editors, *AAAI/IAAI*, pages 314–319. AAAI Press / The MIT Press, 2002.

[24] Vincent Conitzer and Tuomas Sandholm. Common voting rules as maximum likelihood estimators. In *Proceedings of UAI-2005*, 2005.

[25] Stephen A. Cook. The complexity of theorem proving procedures. In *Proceedings of the Third Annual ACM Symposium*, pages 151–158, New York, 1971. ACM.

[26] John Cullinan, SamuelK. Hsiao, and David Polett. A Borda count for partially ordered ballots. *Social Choice and Welfare*, pages 1–14, 2013.

[27] Jessica Davies, George Katsirelos, Nina Narodytska, and Toby Walsh. Complexity of and algorithms for borda manipulation. In *Proceedings of AAAI-2011*, 2011.

[28] Jeffrey Dean and Sanjay Ghemawat. MapReduce: simplified data processing on large clusters. *Communications of the ACM*, 51(1):107–113, 2008.

[29] John Duggan and Thomas Schwartz. Strategic manipulability without resoluteness or shared beliefs: Gibbard-satterthwaite generalized. *Social Choice and Welfare*, 17(1):85–93, 2000.

[30] Cynthia Dwork, Ravi Kumar, Moni Naor, and D Sivakumar. Rank aggregation methods for the Web. In *Proceedings of the 10th international conference on World Wide Web*, pages 613–622, New York, NY, USA, 2001.

[31] David Easley and Jon Kleinberg. *Networks, Crowds, and Markets: Reasoning About a Highly Connected World*. Cambridge University Press, 2010.

[32] Edith Elkind, Piotr Faliszewski, and Arkadii M. Slinko. Cloning in elections: Finding the possible winners. *J. Artif. Intell. Res. (JAIR)*, 42:529–573, 2011.

[33] Ulle Endriss. Judgment aggregation. In Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel Procaccia, editors, *Handbook of Computational Social Choice*. Cambridge University Press, 2015.

[34] Ulle Endriss and Raquel Fernández. Collective annotation of linguistic resources: Basic principles and a formal model. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL-2013)*, August 2013.

[35] Gábor Erdélyi, Michael R. Fellows, Jörg Rothe, and Lena Schend. Control complexity in bucklin and fallback voting: A theoretical analysis. *Journal of Computer and System Sciences*, 81(4):632–660, 2015.

[36] Gábor Erdélyi, Michael R. Fellows, Jörg Rothe, and Lena Schend. Control complexity in bucklin and fallback voting: An experimental analysis. *Journal of Computer and System Sciences*, 81(4):661–670, 2015.

[37] Andrea Esuli and Fabrizio Sebastiani. Sentiment quantification. *IEEE Intelligent Systems*, 25(4):72–75, 2010.

[38] Ronald Fagin, Ravi Kumar, Mohammad Mahdian, D. Sivakumar, and Erik Vee. Comparing partial rankings. *SIAM Journal on Discrete Mathematics*, 20:47–58, 2006.

[39] Piotr Faliszewski, Edith Hemaspaandra, and Lane A. Hemaspaandra. How hard is bribery in elections? *J. Artif. Intell. Res. (JAIR)*, 35:485–532, 2009.

[40] Piotr Faliszewski, Edith Hemaspaandra, and Lane A. Hemaspaandra. Using complexity to protect elections. *Commun. ACM*, 53(11):74–82, 2010.

[41] Piotr Faliszewski, Edith Hemaspaandra, and Lane A. Hemaspaandra. Multimode control attacks on elections. *J. Artif. Intell. Res. (JAIR)*, 40:305–351, 2011.

[42] Piotr Faliszewski, Edith Hemaspaandra, Lane A. Hemaspaandra, and Jörg Rothe. Llull and copeland voting computationally resist bribery and constructive control. *J. Artif. Intell. Res. (JAIR)*, 35:275–341, 2009.

[43] Piotr Faliszewski, Edith Hemaspaandra, LaneA. Hemaspaandra, and Jörg Rothe. A richer understanding of the complexity of election systems. In S.S. Ravi and SandeepK. Shukla, editors, *Fundamental Problems in Computing*, pages 375–406. Springer Netherlands, 2009.

[44] Piotr Faliszewski and Ariel D. Procaccia. AI's war on manipulation: Are we winning? *AI Magazine*, 31(4):53–64, 2010.

[45] Peter C. Fishburn. Paradoxes of voting. *The American Political Science Review*, 68(2):pp. 537–546, 1974.

[46] Murthy Ganapathibhotla and Bing Liu. Mining opinions in comparative sentences. In *Proceedings of COLING-2008*, 2008.

[47] Michael R. Garey and David S. Johnson. *Computers and intractability: a guide to the theory of NP-completeness*. W.J. Freeman and Company, San Francisco, CA, 1979.

[48] Ashutosh Garg, T. S. Jayram, Shivakumar Vaithyanathan, and Huaiyu Zhu. Generalized opinion pooling. In *Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications*, 2004.

[49] Allan Gibbard. Manipulation of voting schemes: A general result. *Econometrica*, 41(4):587–601, 1973.

[50] Jorge Godoy, Dominique Gruyer, Alain Lambert, and Jorge Villagra. Development of an particle swarm algorithm for vehicle localization. In *Intelligent Vehicles Symposium*, pages 1114–1119. IEEE, 2012.

[51] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, MA, 1989.

[52] Umberto Grandi, Andrea Loreggia, Francesca Rossi, and Vijay Saraswat. From sentiment analysis to preference aggregation. In *Proceedings of the International Symposium on Artificial Intelligence and Mathematics (ISAIM-2014)*, 2014.

[53] Umberto Grandi, Andrea Loreggia, Francesca Rossi, Kristen Brent Venable, and Toby Walsh. Restricted manipulation in iterative voting: Condorcet efficiency and borda score. In Patrice Perny, Marc Pirlot, and Alexis Tsoukiàs, editors, *ADT*, volume 8176 of *Lecture Notes in Computer Science*, pages 181–192. Springer, 2013.

[54] Mingyu Guo and Vincent Conitzer. Computationally feasible automated mechanism design: General approach and case studies. In Maria Fox and David Poole, editors, *AAAI*. AAAI Press, 2010.

[55] Edith Hemaspaandra, Lane A. Hemaspaandra, and Jörg Rothe. Anyone but him: The complexity of precluding an alternative. *Artif. Intell.*, 171(5-6):255–285, 2007.

[56] Mehran Hojati. Optimal political districting. *Computers OR*, 23(12):1147–1161, 1996.

[57] Matthew O. Jackson. *Social and economic networks*. Princeton University Press, Princeton, NJ, 2008.

[58] Nitin Jindal and Bing Liu. Mining comparative sentences and relations. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI-2006)*, 2006.

[59] Toshihiro Kamishima, Hideto Kazawa, and Shotaro Akaho. A survey and empirical comparison of object ranking methods. In Johannes Fürnkranz and Eyke Hüllermeier, editors, *Preference Learning*, pages 181–201. Springer-Verlag, 2010.

[60] Richard M. Karp. Reducibility among combinatorial problems. In R.E. Miller and J.W. Thatcher, editors, *Complexity of Computer Computations*. Plenum Press, New York, 1972.

[61] James Kennedy and Russell C. Eberhart. Particle swarm optimization. In *Proceedings of the IEEE International Conference on Neural Networks*, pages 1942–1948, 1995.

[62] Jérôme Lang and Lirong Xia. Voting in combinatorial domains. In Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel Procaccia, editors, *Handbook of Computational Social Choice*. Cambridge University Press, 2015.

[63] Jérôme Lang, Nicolas Maudet, and Maria Polukarov. New results on equilibria in strategic candidacy. In Berthold Vöcking, editor, *SAGT*, volume 8146 of *Lecture Notes in Computer Science*, pages 13–25. Springer, 2013.

[64] Jean-François Laslier and M. Remzi Sanver, editors. *Handbook of Approval Voting*. Springer, 2010.

[65] Omer Lev and Jeffrey S. Rosenschein. Convergence of iterative voting. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS-2012)*, 2012.

[66] Andrew Lin. The complexity of manipulating k-approval elections. In Joaquim Filipe and Ana L. N. Fred, editors, *ICAART (2)*, pages 212–218. SciTePress, 2011.

[67] Bing Liu. *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data (Data-Centric Systems and Applications)*. Springer-Verlag New York, Inc., 2006.

[68] Bing Liu. *Sentiment Analysis and Opinion Mining*. Synthesis Lectures on Human Language Technologies. Morgan and Claypool Publishers, 2012.

[69] Andrea Loreggia. Iterative voting and multi-mode control in preference aggregation. Master's thesis, University of Padova, 2012.

[70] Andrea Loreggia, Nina Narodytska, Francesca Rossi, K. Brent Venable, and Toby Walsh. Controlling elections by replacing candidates or votes. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '15, pages 1737–1738, Richland, SC, 2015. International Foundation for Autonomous Agents and Multiagent Systems.

[71] Andrew Mao, Ariel D. Procaccia, and Yiling Chen. Better human computation through principled voting. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence (AAAI-2013)*, 2013.

[72] Nicholas Mattei and Toby Walsh. Preflib: A library for preferences http://www.preflib.org. In Patrice Perny, Marc Pirlot, and Alexis Tsoukiàs, editors, *ADT*, volume 8176 of *Lecture Notes in Computer Science*, pages 259–270. Springer, 2013.

[73] Nicolas Maudet, Jérôme Lang, Yann Chevaleyre, and Guillaume Ravilly-Abadie. Compiling the votes of a subelectorate. In *Proceedings of IJCAI-2009*, 2009.

[74] Reshef Meir, Maria Polukarov, Jeffrey S. Rosenschein, and Nicholas R. Jennings. Convergence to equilibria in plurality voting. In *Proceedings of the Twenty-fourth conference on Artificial Intelligence (AAAI-2010)*, 2010.

[75] Curtis Menton. Normalized range voting broadly resists control. *Theory Comput. Syst.*, 53(4):507–531, 2013.

[76] P.T. Metaxas, E. Mustafaraj, and D. Gayo-Avello. How (not) to predict elections. In *Proceedings of PASSAT-2011 and SOCIALCOM-2011*, 2011.

[77] Gilad Mishne. Predicting movie sales from blogger sentiment. In *In AAAI 2006 Spring Symposium on Computational Approaches to Analysing Weblogs (AAAI-CAAW)*, 2006.

[78] Hervé Moulin. Condorcet's principle implies the no show paradox. *Journal of Economic Theory*, 45(1):53–64, 1988.

[79] Lihi Naamani Dery, Meir Kalech, Lior Rokach, and Bracha Shapira. Iterative voting under uncertainty for group recommender systems. In *Proceedings of the fourth ACM conference on Recommender systems (RECSYS-2010)*, 2010.

[80] Nina Narodytska and Toby Walsh. Manipulating two stage voting rules. In *Proceedings of the Twelft International conference on Autonomous Agents and Multi-Agent Systems (AAMAS-2013)*, 2013.

[81] Svetlana Obraztsova and Edith Elkind. Optimal manipulation of voting rules. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS-2012)*, 2012.

[82] Svetlana Obraztsova, Evangelos Markakis, Maria Polukarov, Zinovi Rabinovich, and Nick Jennings. On the convergence of iterative voting: How restrictive should restricted dynamics be? In *Proceedings of the 5th International Workshop on Computational Social Choice (COMSOC-2014)*, 2014.

[83] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and Trends in Information Retrieval*, 2(1-2):1–135, 2008.

[84] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of EMNLP-2002*, pages 79–86, 2002.

[85] Maria Silvia Pini, Francesca Rossi, Kristen Brent Venable, and Toby Walsh. Incompleteness and incomparability in preference aggregation: Complexity results. *Artificial Intelligence*, 175(7-8):1272–1289, 2011.

[86] Zinovi Rabinovich, Svetlana Obraztsova, Omer Lev, Evangelos Markakis, and Jeffrey Rosenschein. Analysis of equilibria in iterative voting schemes. In *Proceedings of the 5th International Workshop on Computational Social Choice (COMSOC-2014)*, 2014.

[87] Annemieke Reijngoud and Ulle Endriss. Voter response to iterated poll information. In *Proceedings of the 11th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2012)*, June 2012.

[88] Reyhaneh Reyhani and Mark C. Wilson. Best reply dynamics for scoring rules. In *Proceedings of the 20th European Conference on Artificial Intelligence (ECAI-2012)*, pages 672–677, 2012.

[89] Federica Ricca, Andrea Scozzari, and Bruno Simeone. Political districting: from classical models to recent approaches. *4OR*, 9(3):223–254, 2011.

[90] Francesca Rossi, Kristen Brent Venable, and Toby Walsh. *A Short Introduction to Preferences: Between Artificial Intelligence and Social Choice*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan and Claypool Publishers, 2011.

[91] Nathan F. Russell. Complexity of Control of Borda Count Elections, 2007.

[92] Mark Allen Satterthwaite. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10(2):187 – 217, 1975.

[93] Yoav Shoham and Kevin Leyton-Brown. *Multiagent Systems - Algorithmic, Game-Theoretic, and Logical Foundations.* Cambridge University Press, 2009.

[94] Piotr Skowron, Piotr Faliszewski, and Arkadii Slinko. Achieving fully proportional representation is easy in practice. In *In Proc. of the 2013 AAMAS*, AAMAS '13, pages 399–406, Richland, SC, 2013. International Foundation for Autonomous Agents and Multiagent Systems.

[95] Sydney Coordinated Adaptive Traffic System. SCATS: How it works | Adaptive control. `http://www.scats.com.au/how-scats-works-adaptive.html`, 2014.

[96] Alan D. Taylor. *Social choice and the mathematics of manipulation.* Cambridge University Press, 2005.

[97] David R. M. Thompson, Omer Lev, Kevin Leyton-Brown, and Jeffrey S. Rosenschein. Empirical analysis of plurality election equilibria. In *The Twelfth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2013)*, 2013.

[98] Frank van Harmelen, Frank van Harmelen, Vladimir Lifschitz, and Bruce Porter. *Handbook of Knowledge Representation.* Elsevier, 2007.

[99] Toby Walsh. An empirical study of the manipulability of single transferable voting. In Helder Coelho, Rudi Studer, and Michael Wooldridge, editors, *ECAI*, volume 215 of *Frontiers in Artificial Intelligence and Applications*, pages 257–262. IOS Press, 2010.

[100] Toby Walsh. Is computational complexity a barrier to manipulation? *Annals of Mathematics and Artificial Intelligence*, 62(1-2):7–26, June 2011.

[101] Lirong Xia and Vincent Conitzer. A maximum likelihood approach towards aggregating partial orders. In *Proceedings of IJCAI-2011*, 2011.

[102] Lirong Xia, Michael Zuckerman, Ariel D. Procaccia, Vincent Conitzer, and Jeffrey S. Rosenschein. Complexity of unweighted coalitional manipulation under some common voting rules. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI-2009)*, 2009.

[103] H. Peyton Young. An axiomatization of Borda's rule. *Journal of Economic Theory*, 9(1):43–52, 1974.

List of Algorithms

1 Destructive Control Replacing Votes 32

2 Constructive Control Replacing Votes 34

3 Constructive Control Replacing Voters 36

4 Destructive Control Replacing Candidates 44

5 Computing winner 64

6 Computing BU 65

List of Figures

3.1 Sushi dataset: fraction of profiles (over 1000) with successful DCRC. . . . 47

3.2 T-shirt dataset: fraction of profiles (over 1000) with successful DCRC. . . . 48

3.3 Deterministic algorithm on sushi dataset: average percentage of used combinations. 49

3.4 Deterministic algorithm on t-shirt dataset: average percentage of tried combinations. 49

3.5 Deterministic and probabilistic algorithm comparison on the sushi dataset. 50

3.6 Deterministic and probabilistic algorithm comparison on the t-shirt dataset. 50

3.7 Deterministic algorithm on sushi dataset using plurality: RC compared to AC and DC. 52

3.8 Deterministic algorithm on sushi dataset using veto: RC compared to AC and DC. 52

3.9 Deterministic algorithm: RC compared to AC, DC, and AC+DC. 53

3.10 Borda deterministic and probabilistic algorithm: comparison. 54

3.11 Borda: percentage of profiles (over 1000) with successful DCRC. 54

4.1 Copeland does not converge with best response manipulation. 62

4.2 STV with sequential turn function does not converge. 68

4.3 A dominated candidate elected by PSR with scores (2,1,1,1,0). 70

4.4 Number of profiles with iteration compared to the correlation ratio. . . . 73

4.5 Doodle experiment with UM10: Condorcet efficiency. 73
4.6 Electoral experiment with UM10: Condorcet efficiency. 74
4.7 Sushi Dataset experiment: Condorcet efficiency 75

5.1 SP-structures associated with Example 5.2. 87
5.2 Percentage of profiles where sentiment analysis and $B^*\underline{\alpha}$ differ (2 candidates). 100
5.3 Percentage of profiles where sentiment analysis and $B^*\underline{\alpha}$ differ (10 voters). 101
5.4 Mean error of $B^*\underline{\alpha}$ on incomplete profiles. 102