# From Elections to Tournaments: A Study of the Computational Complexity of Bribery, Design, and Prediction Problems in Voting and Sports

Inaugural-Dissertation

zur Erlangung des Doktorgrades
der Mathematisch-Naturwissenschaftlichen Fakultät
der Heinrich-Heine-Universität Düsseldorf


vorgelegt von

Tobias Alexander Hogrebe

geboren in Leverkusen


Düsseldorf, Mai 2022

aus dem Institut für Informatik
der Heinrich-Heine-Universität Düsseldorf

Gedruckt mit der Genehmigung der
Mathematisch-Naturwissenschaftlichen Fakultät der
Heinrich-Heine-Universität Düsseldorf

Berichterstatter:

1. Jun.-Prof. Dr. Dorothea Baumeister

2. Dr. Umberto Grandi

3. Prof. Dr. Davide Grossi

Tag der mündlichen Prüfung: 9. Dezember 2022

# Abstract

This thesis is concerned with the study of the computational complexity of problems related to elections and sport tournaments from the field of computational social choice. The latter field, which is one of the youngest areas of theoretical computer science, primarily builds on the field of computational complexity theory, which studies the difficulty of solving problems from an algorithmic perspective, and the field of social choice theory, which studies decision-making processes from an axiomatic perspective. At its heart lies the study of the computational complexity of problems related to decision-making processes such as elections, tournaments, resource allocation, judgment aggregation, and matching. Despite its relatively young age, computational social choice is considered as a central and very active field of artificial intelligence and multi-agent systems research, attracting computer scientists, economists, and sociologists alike.

A key aspect that accompanies us through this thesis is the concept of uncertainty in elections and tournaments in relation to computational complexity. This includes, for example, uncertainty about the preferences of voters, about the voting rule, and about the outcome of the remaining matches in a tournament. In more detail, we focus on the following topics.

We start by studying the computational complexity of decision problems concerning bribery and the evaluation of the robustness of election outcomes. We apply concepts from classical decision complexity and show that the problems can be solved in polynomial time for certain combinations of voting rules and types of functions measuring the strength of changes or uncertainty in the votes, while for other cases they are NP-complete and thus unlikely to have polynomial-time solutions. Afterwards, we study the computational complexity of the problem of designing scoring systems for elections and competitions with the goal of guaranteeing the victory of a desired candidate or checking the robustness of a candidate's victory with respect to the system used. Besides various results regarding the classical decision complexity, in terms of membership in P and NP-completeness, we further differentiate the complexity with respect to different parameterizations and show FPT and W[2]-hardness results and conclude with experiments on real-world data. Regarding elections with probabilistically distributed preferences, we study the function complexity of determining the winning probabilities of candidates for different combinations of voting rules, distribution models, tie-breaking procedures, and parameterizations using concepts from counting complexity. We show membership in FP for some cases, while showing #P-hardness for other cases. We then move to round-robin tournaments and study the function complexity of calculating the probability that a given team ends up as the champion under the assumption that we are in the course of a tournament, whereby some of the matches have already been played while others remain to be played. Again, we apply concepts of counting complexity, examine different parameterizations, and show memberships in FP, #P-hardness, and FPT results. In addition, we perform experiments on real-world data and synthetic data and, motivated by the empirical results, study the average-case complexity of the problem and show the expected polynomial time of our FPT algorithm for certain distributions.

Finally, we switch from studying the complexity of problems whose instances model uncertainties to studying the complexity of problems under the assumption of uncertainties about the instances and discuss our proposal of applying the concept of smoothed analysis in the field of computational social choice.

# Contents

# Chapter 1

## Introduction

Elections and tournaments are central mechanisms of coexistence and peaceful competition. The use of elections ranges from the election of parliaments and presidents of entire countries to the election of the spokesperson of a small group or the administration of a local club. Similarly, tournaments range from world cups, the Olympics, and major national sports leagues to local tournaments and school competitions. At their core, elections and tournaments have the same goal: to select the most eligible candidates based on certain criteria in a fair manner. However, the problem of deciding what these criteria are and what is meant by in a fair manner is far from trivial. The scientific and formal study of these questions are central tasks of *social choice theory* and related fields.

The formal study of elections from a mathematical point of view was most notably initiated in the 18th century by Jean-Charles de Borda and Marie Jean Antoine Nicolas de Caritat, the Marquis de Condorcet. Borda proposed a new voting rule in which each voter provides a ranking over the candidates, according to which the candidates receive as many points as there are candidates ranked below. In the end, the candidates with the highest score are presented as the winners. He argued that his voting rule, which we refer to as the Borda voting rule nowadays, is superior to plurality voting, in which each voter can award only a single point to his or her preferred candidate, because it would take into account the pairwise comparisons between the candidates. Condorcet, on the other hand, argued that there are scenarios in which a candidate exists who beats every other candidate in a direct pairwise majority comparison, but would still not be determined as the winner by the Borda voting rule. However, the consideration of direct pairwise majority comparisons can end up in a bizarre situation where circles are formed, as we will see in the following example. Consider the following election with candidates $a$, $b$, and $c$ and three voters. Voter 1 ranks candidate $a$ ahead of $b$ followed by $c$. Voter 2 ranks candidate $c$ before $a$ followed by $b$. Finally, voter 3 ranks candidate $b$ before $c$ followed by $a$. Candidate $a$ beats candidate $b$ in a direct pairwise majority comparison as voter 1 and voter 3 prefer $a$ over $b$. Candidate $b$ beats $c$ as voter 1 and voter 2 prefer $b$ over $c$. Eventually, however, $c$ also beats $a$ as voter 2 and voter 3 prefer $c$ over $a$. Thus, for each candidate there is another candidate who beats him or her in a direct pairwise majority comparison. This phenomenon is known as the *Condorcet paradox* and it shows that not in every scenario a winner can be determined using only direct pairwise majority comparisons. Thus, in the following decades, an ongoing endeavor began to discover new voting rules and to study their properties. Fundamental to the modern orientation of social choice was the work of Nobel laureate Kenneth Arrow [1951], who showed, among other things, that certain intuitively reasonable properties of voting rules are incompatible, in the sense that no rule exists which satisfies them simultaneously. This result is commonly known as *Arrow's impossibility theorem*. Later, Gibbard [1973] and Satterthwaite [1975] independently showed that no voting rule that meets a certain set of properties, which may be seen as minimum requirements a meaningful voting rule should fulfill, can be immune to manipulation by untruthful voting. For a comprehensive overview on the history of social choice theory, we refer the reader to Urken [1995].

Following the results of Gibbard and Satterthwaite, Bartholdi, Tovey, and Trick [1989a] studied the algorithmic complexity, in other words, the amount of computational resources required, of manipulating elections. They wanted to evaluate whether the possible computational burden of performing a manipulation could provide some level of protection. The field that deals with the complexity of solving problems is *computational complexity theory*, based on the fundamental work of Alan Turing in the first half of the last century. Thus, the work of Bartholdi, Tovey, and Trick marked the emergence of the field to which most of our work here can be assigned, the field of *computational social choice*. It arises from the intersection of the aforementioned fields of social choice theory and computational complexity theory and is especially concerned with the study of problems related to decision making processes from, but not only, a computational point of view. Note, however, that the topics studied by computational social choice are not limited to elections. The focus is also on *tournaments*, *resource allocation* including the *allocation of indivisible goods* and *cake cutting*, *matching* under preferences, *hedonic games*, *judgment aggregation*, and many other topics. Moreover, it has become evident that the connection between computational social choice and related fields is bidirectional, as many advances in the area of computational social choice carry back to social choice theory and computational complexity theory, see, e.g., Hemaspaandra [2018].

The developments in recent decades, such as the digitization of elections, the introduction of direct democratic participation processes such as participatory budgeting, the establishment of social networks, and the ever-increasing acceptance and spread of online games and e-sports, have resulted in a high amount of recurring tasks involving large amounts of data and new vulnerabilities. Thus, the question of the computational complexity of solving and protecting these tasks has never been as relevant as it is today.

In this thesis we will consider a number of various known and novel problems and present substantial contributions to the study of their computational complexity. In the first part of the thesis, we focus in particular on bribery, voting rule design, robustness, and prediction problems arising in elections, with a special focus on problems related to distance measures. Our study of election prediction problems then leads into the second part of the thesis, where we switch from elections to tournaments and study prediction problems in the context of round-robin tournaments. We use a variety of different concepts from computational complexity theory to examine the problems. For instance, in addition to classical decision complexity, we also consider counting complexity, parameterized complexity, approximation, average-case complexity, and empirical analysis using experiments on real-world data and synthetic data. In addition, we also discuss the potential applications of smoothed analysis in the area of computational social choice. We successfully apply a variety of different solution approaches such as classical deterministic algorithms, heuristics and brute-force algorithms, dynamic programming, (minimum cost) network flow, and linear and non-linear optimization.

Thus, a key concept that will recur throughout this thesis is uncertainty in relation to computational complexity. This includes studying problems that model uncertainty in their instances, for example, by uncertainty about the preferences of the voters, about the voting rule, or about the outcomes of the remaining matches in a tournament, as well as studying the complexity of problems assuming certain degrees of uncertainty over the instances themselves using average-case analysis and smoothed analysis.

## Outline

The remaining thesis is structured as follows. In Chapter 2, we briefly present the foundations and backgrounds of computational complexity theory and computational social choice that we employ. In particular, we will discuss the literature and provide a context for our research. In the following chapters we then illustrate and discuss our contributions to the field, and in particular address the respective impact of the contribution and the related work.

In Chapter 3 we study the problem of priced bribery in elections in the constructive and destructive cases and for various different distance measures with varying degrees of variability and expressiveness. We study the classical decision complexity of the problems with respect to well-known voting rules and also develop dichotomy results for the class of scoring rules.

In Chapter 4 we introduce the problem of designing scoring rules for elections and scoring systems for competitions over rankings, such as racing competitions or the Eurovision Song Contest, that guarantee the victory of a particular candidate. Furthermore, we investigate the extension in which a scoring system is already in place and we try to modify the system as little as possible with respect to different distance measures. In addition to the classical decision complexity, we also study the parameterized complexity with respect to various natural parameters. Our theoretical results are complemented by experiments on real-world data to examine the complexity and relevance of the problem in practice.

In Chapter 5 we study the evaluation problem which is the problem of determining the winning probability of candidates in elections in which the votes are distributed probabilistically. We study the function complexity of the problem using notions from counting complexity and parameterized counting complexity. We consider different scoring rules, three different types of distribution models, tie-breaking procedures, and the decision variant. In particular, we discuss the connection to the results obtained in Chapter 3 and discuss different motivations such as election prediction and the evaluation of the robustness of election outcomes.

In Chapter 6, we then move from elections to round-robin tournaments, as used for example in many major national sports leagues. We study the worst-case complexity, the parameterized complexity, and also the average-case complexity of the evaluation problem, i.e., the determination of the probability of a given team to end up as the champion, and also of the well-known sports elimination problem, in which we as whether a given team can still become the champion. The theoretical results are complemented by experiments on real-world data and synthetic data.

In Chapter 7, we then discuss our proposal on the possibilities of applying the smoothed analysis of Spielman and Teng [2004, 2009] in the area of computational social choice. In particular, we will discuss the subsequent related work.

Finally, in Chapter 8, we will summarize our results, discuss them as a whole, and give an outlook on possible future work and directions.

# Chapter 2

## Background

This chapter deals with the foundations and backgrounds of computational complexity theory and computational social choice relevant to our research presented here.

## 2.1 Computational Complexity

At the heart of computational complexity lies the question of how hard it is to solve problems from a computational point of view. These problems can have diverse shapes, for example they can be about computing mathematical functions, breaking cryptographic encryptions, or solving puzzles. In the following, we will give a short introduction to the basics of computational complexity relevant for this work. For a comprehensive overview we refer to the books of Papadimitriou [1994] and Arora and Barak [2009], which we have also used as a guideline here.

### 2.1.1 Decision Complexity

In the first half of the last century, Turing [1937b] laid the foundations for the theory of computation in his seminal paper. His goal was to study the computability of the *Entscheidungsproblem* (German for decision problem) proposed by Hilbert and Ackermann [1928], which asks whether an algorithm exists that receives formal statements as input and outputs whether they are true or false. For this he introduced the concept of the so-called *Turing machine*, a finite rule-based automaton with access to a set of linear tapes, which he used to capture the concept of computability. We have illustrated the basic layout of a Turing machine in Figure 2.1. In fact, the generally accepted Church–Turing thesis states that the class of intuitively computable functions corresponds to the class of functions computable by Turing machines. This assumption is based on the work by Kleene [1936], Church [1936], and Turing [1937a] who showed that the existing concepts of intuitive computability, namely *general recursivity* (Gödel [1934], based on suggestions by Jacques Herbrand), $\lambda$-*calculus* (Church [1936]), and *Turing computability* (Turing [1937b]) are all equivalent. Thus, for convenience, and because it is beneficial for practical applicability, we often describe Turing machines by their specification in the form of an *algorithm* in pseudocode or descriptive language and often use the terms Turing machine and algorithm interchangeably. Note that unless stated otherwise, by Turing machine we refer to a deterministic Turing machine.

#### Decision Problems

As in the original *Entscheidungsproblem* by Hilbert and Ackermann [1928], we start by focusing on problems for which the answer can take only two different values: `yes` or `no`. This type

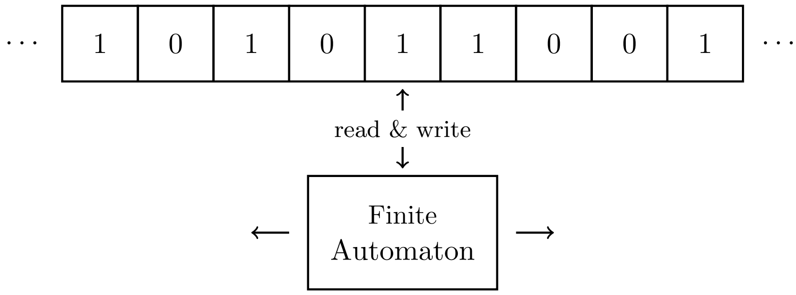


Figure 2.1: Depiction of a Turing machine with access to a single tape over the alphabet $\Sigma = \{0, 1\}$ consisting of a finite automaton that determines the symbol to be written and the movement to be performed based on the current state and the symbol being read.

of problems is referred to as decision problems. In the context of computational complexity, arguably the best-known decision problem is the *Boolean satisfiability problem.*

| Satisfiability (SAT) | |
|---|---|
| **Given:** | A Boolean formula $\phi$ over a set $X = \{x_1, \ldots, x_n\}$ of Boolean variables. |
| **Question:** | Does there exist an assignment of the variables in $X$ such that $\phi$ is satisfied? |

We refer to the input of a decision problem as an *instance*. So in the case of SAT, an instance consists of a certain formula over a certain finite set of variables. We call such an instance of a problem a *yes-instance* if the answer for it is `yes`. Accordingly, we call the instance a *no-instance* if the answer for it is `no`.

**Example 2.1.** Consider the following instance of SAT consisting of the formula

$$\phi = (x_1 \vee x_2 \vee x_4) \wedge (\neg x_2 \vee \neg x_4) \wedge (\neg x_1 \vee x_2) \wedge (\neg x_1 \vee x_3)$$

over the set of Boolean variables $X = \{x_1, x_2, x_3, x_4\}$. An assignment setting the formula $\phi$ to *true* is $x_1 = false$, $x_2 = false$, $x_3 =$ *true* and $x_4 =$ *true*. Thus, the answer to the question of the problem, whether an assignment exists for which the formula becomes true, is `yes` whereby the given instance is a yes-instance.

### Time Complexity

While it was quite simple to check whether there exists an assignment which sets the formula in the previous example to true, this can become considerably more difficult for formulas over a larger number of variables. For 8 variables there are $2^8 = 256$ combinations, which we can check with enough time by hand and quite fast with a modern computer to see if a suitable assignment exists. For 20 variables, however, there are already $2^{20} = 1048576$ combinations, for which we, even if we only need 10 seconds per combination by hand, already need just over 121 days to check and should therefore better use a computer, for which even such a high number can be managed in a few seconds. But what happens if we want to do the same for 100 variables? Even if our modern computer could check $10^{12}$ assignments per second, it

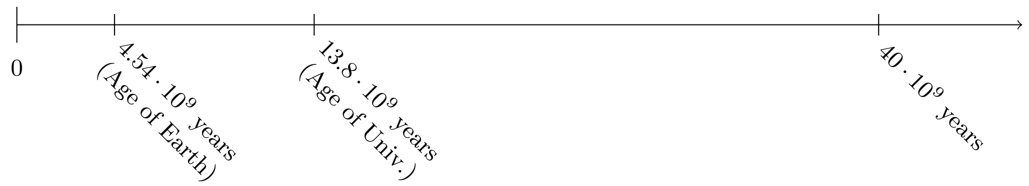


Figure 2.2: Time scale to visualize the length of the exemplary running time of the brute force algorithm for SAT compared to estimated real-world time durations.

would need just over 40 billion years to check all $2^{100} \approx 1.27 \cdot 10^{30}$ assignments, thus only a little bit less than three times the current estimated age of our universe, and just over 533 times the time required by Deep Thought to compute 42, the "Answer to the Ultimate Question of Life, the Universe, and Everything" according to the *The Hitchhiker's Guide to the Galaxy* by Douglas Adams [1979]. See Figure 2.2 for a visual comparison.

The problem here is the exponential growth of the number of possible solutions and the associated exponential growth of time required to check them. Unfortunately, this growth occurs in a wide variety of problems. The question now is whether it is possible to find shortcuts in order to develop faster approaches to solve these problems. This leads us to the field of *computational complexity*.

Based on the strong theoretical groundwork mentioned at the beginning of this section, Hartmanis and Stearns [1965] build the foundation of computational complexity theory by proposing that the complexity of problems can be classified by the number of steps required by a Turing machine to solve them. The classification based on the number of steps required is the so-called *time complexity*. On the other hand, the classification based on the number of tape cells required by a Turing machine is the so-called *space complexity*. The investigation of space complexity was initiated by Myhill [1960] even before the investigation of the time complexity, which is the main focus here.[1] Thus, when we speak about computational complexity in the following, we are referring to time complexity.

As suggested by Hartmanis and Stearns [1965], the time complexity of a Turing machine is measured by how its worst-case runtime behaves asymptotically with respect to the input size. The worst-case running time of a deterministic Turing machine depending on the input size $n = |I|$ is defined as the maximum number of steps the Turing machine needs to decide an instance of the respective size. The central notation for the classification of different growth rates is the Bachmann–Landau notation, and here in particular the so-called Big $\mathcal{O}$ notation. We say that a function $f$ is asymptotically bounded proportional to a function $g$ if $\exists c > 0 : \exists n_0 \in \mathbb{N} : \forall n \geq n_0 : f(n) \leq c \cdot g(n)$ holds, which we denote by $f \in \mathcal{O}(g)$.

We say that a deterministic Turing machine has a polynomial running time if its worst-case running time is bounded by a polynomial. Accordingly, we say that a decision problem can be solved in deterministic polynomial time if there exists a deterministic Turing machine with polynomial running time which solves the problem.

[1]It should be mentioned here that the time complexity also limits the space complexity, since a Turing machine can only write to a new cell on the tape if it also makes a step.

By default, and without loss of generality, we assume that all inputs are encoded in binary, this includes numbers, strings, lists, and more complex objects. For natural numbers, it is also interesting in some scenarios to assume that they are encoded in unary. This means that for the number $k \in \mathbb{N}$, exactly $k$ ones are written to the tape and thus $k$ tape cells are needed instead of $\lfloor \log_2(k) \rfloor + 1$ cells for the binary encoding. Therefore, an algorithm that has an exponential running time depending on the length of the input encoded in binary can have a polynomial running time under the assumption that the input is encoded in unary.

When dealing with the asymptotic worst-case running time analysis as defined above, we have to be careful with respect to the practical efficiency of algorithms. First of all, the running time in the worst-case may be misleading with respect to the running time of the algorithm in practice, where the worst-case instances may not occur at all or not very frequently. This issue will be discussed later with respect to parameterized complexity and average-case complexity. Furthermore, the asymptotic running time analysis neglects constant factors and portions of the running time. For example, an algorithm with a number of $2^{100} \cdot n$ steps in the worst case has a linear running time $\mathcal{O}(n)$, which is usually interpreted as very efficient, while no currently realizable machine could perform this computation in a feasible amount of time even for small inputs.

### Complexity Classes

For a rough classification of problems with respect to their computational difficulty, the problems are often sorted into so-called complexity classes. The two most prominent classes are P and NP. The class P contains all decision problems which can be solved in deterministic polynomial time. Accordingly, the class NP contains all decision problems for which a given *solution* (sometimes referred to as a *witness*) for a yes-instance can be verified in polynomial time.[2] However, while yes-instances for problems in NP can be verified in polynomial time by given solutions, which may themselves have only polynomial length due to the time constraint, this is explicitly not required for no-instances. For example, although it is easy to see that SAT is in NP, since a possible solution for a yes-instance in the form of an assignment can be verified in polynomial time by evaluating the formula with respect to the given assignment, for a no-instance it would have to be verified that none of the exponentially many possible assignments sets the formula to true.

The *complement class* co$\mathcal{C}$ of a complexity class $\mathcal{C}$ consists of the problems in $\mathcal{C}$ with negated questions. Thus, a yes-instance in $\mathcal{C}$ is a no-instance in co$\mathcal{C}$ and vice versa. For example, the coNP counterpart of SAT is $\overline{\text{SAT}}$, which asks if no assignment exists that sets the formula to true. According to the definition, solutions for no-instances for problems in coNP can be verified in polynomial time. Note that P = coP, but the exact relationship between NP and coNP is not clear. However, it is suspected that NP $\neq$ coNP.

It is straightforward to verify that P $\subseteq$ NP holds. However, whether P = NP or P $\neq$ NP is true, is a long-standing open question. In the following we will discuss how the relationship

[2] As the name NP, which stands for non-deterministic polynomial time, suggests, the original definition of the class is not the one presented here, but the equivalent definition over the length of the shortest accepting path(s) of non-deterministic Turing machines.

between the complexity of problems is studied and why it is commonly assumed that P $\neq$ NP and therefore P $\subsetneq$ NP holds.

Suppose we are given two decision problems $A$ and $B$. If we notice that we can solve problem $A$ quite easily by solving problem $B$, we can draw the following conclusion: Problem $B$ seems to be at least as hard as problem $A$. In fact, it is apparent that if we have an approach to problem $B$, we can also solve problem $A$, and, on the other hand, if problem $A$ is very difficult to solve, problem $B$ should also be very difficult to solve. This idea of comparing two problems is called *reduction* and is a key concept of computational complexity.

The most prominent type of reductions for decision problems in the literature and the one we use here is the *polynomial-time many-one reduction*. We say that a decision problem $A$ is polynomial-time many-one reducible to decision problem $B$, denoted by $A \leq_m^p B$, if there exists a Turing machine which transforms each instance $I$ of problem $A$ in polynomial time into an instance $I'$ of problem $B$, where $I$ is a yes-instance of $A$ if and only if $I'$ is also a yes-instance of $B$. We refer to a decision problem $B$ as *$\mathcal{C}$-hard* for a complexity class $\mathcal{C}$ with respect to the polynomial-time many-one reduction if $A \leq_m^p B$ holds for every $A \in \mathcal{C}$. We refer to a decision problem $B$ as *$\mathcal{C}$-complete* for a complexity class $\mathcal{C}$ with respect to the polynomial-time many-one reduction if $B$ is $\mathcal{C}$-hard and $B \in \mathcal{C}$. A very convenient property that the polynomial-time many-one reduction, and also the reductions we will consider later, satisfy is *transitivity*, which means that if $A \leq_m^p B$ and $B \leq_m^p C$ hold, it follows that $A \leq_m^p C$.

Referring back to our observations from the previous informal motivation of reductions, we see here that, for two problems $A$ and $B$ with $A \leq_m^p B$, that if there exists a polynomial-time algorithm for $B$, the combination of that with the polynomial-time algorithm used in the reduction yields a polynomial-time algorithm for $A$. On the other hand, this conclusion also means that if we assume that no polynomial-time algorithm exists for $A$, neither should there be one for $B$. This is where we return to SAT and why it is one of the central problems of computational complexity. Cook [1971] showed that every problem in NP is polynomial-time many-one reducible to SAT and thus that SAT is NP-complete. Therefore, if SAT could be solved in polynomial time, every problem in NP would be solvable in polynomial time, whereby P = NP would follow. However, since decades of various approaches have failed to find a polynomial-time algorithm for SAT, the common assumption is that such an algorithm does not exist for SAT and thus also not for every other NP-hard problem. Building on Cook's theorem, Karp [1972] proved the NP-completeness of twenty-one further decision problems, such as the clique problem, the feedback arc set problem, the graph coloring problem, and the 3-SAT problem, by reducing SAT to them. Here he also established the nowadays common approach to show the NP-hardness of problems not each time via a direct reduction from SAT, but, based on the previously mentioned transitivity of the polynomial-time many-one reduction, by a reduction from other problems already proven to be NP-hard. An example for such a chain of reductions is the proof by Karp [1972] of the NP-hardness of the feedback arc set problem by showing that SAT can be reduced to the clique problem, the clique problem to the vertex cover problem, and the vertex cover problem to the feedback arc set problem. Later, the NP-completeness of many of the problems used in the literature was surveyed, and in some cases also shown, by Garey and Johnson [1979]. In Figure 2.3 we have illustrated the reduction chains up to some of the problems shown to be NP-hard in this work.

The classes P and NP are themselves only the lowest level of a larger structure of complexity classes called the *polynomial-time hierarchy* introduced by Meyer and Stockmeyer [1972].

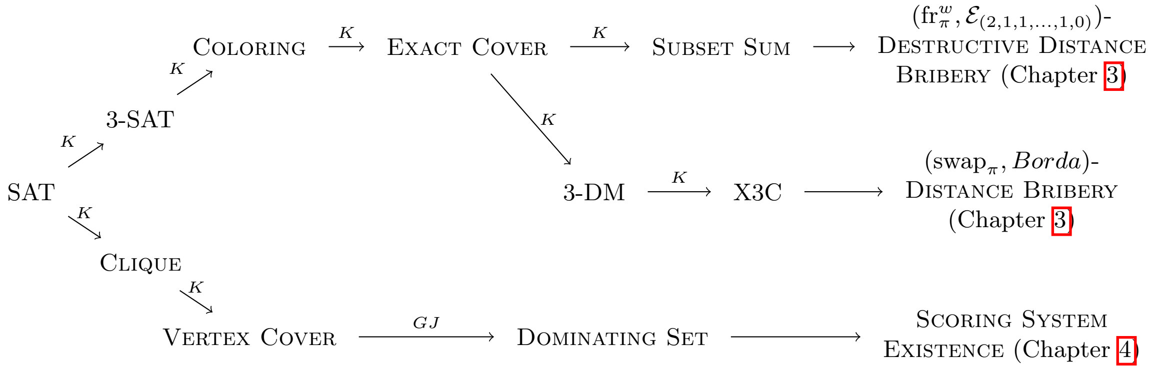


Figure 2.3: Examples of chains of polynomial-time many-one reductions to prove the NP-hardness of problems in this work. Each edge indicates a reduction from left to right. Edges marked with $K$ indicate reductions given by Karp [1972], thus $GJ$ marks reductions given, or hinted, by Garey and Johnson [1979].

Each level consists of three classes defined by $\Sigma_i^p = \mathrm{NP}^{\Sigma_{i-1}^p}$, $\Delta_i^p = \mathrm{P}^{\Sigma_{i-1}^p}$, and $\Pi_i^p = \mathrm{co}\Sigma_i^p$ for $i \geq 1$ and $\Delta_0^p = \Sigma_0^p = \Pi_0^p = \mathrm{P}$. Here $\mathrm{P}^{\Sigma_{i-1}^p}$ means that the problems of that class can be solved in polynomial time by a deterministic Turing machine with access to a $\Sigma_{i-1}^p$ oracle, that is a black box machine which returns answers to problems in $\Sigma_{i-1}^p$ for given instances in constant time. Likewise, the problems of $\mathrm{NP}^{\Sigma_{i-1}^p}$ can be solved by a non-deterministic Turing machine with access to an $\Sigma_{i-1}^p$ oracle in polynomial time. For example, $\Delta_2^p = \mathrm{P}^{\mathrm{NP}}$ is the class of decision problems that can be solved in polynomial time by a deterministic Turing machine with access to an NP oracle. Note that all classes of one level of the polynomial-time hierarchy are actually included in every single class of the next level, thus $\Sigma_i^p \cup \Delta_i^p \cup \Pi_i^p \subseteq \Sigma_{i+1}^p$ and vice versa for $\Delta_{i+1}^p$ and $\Pi_{i+1}^p$ for $i \geq 0$. Like P and NP, all classes of PH have their own complete problems with respect to the polynomial-time many-one reduction and, also as for P and NP, it is assumed that the complexities of the classes are truly distinct for $i \geq 1$. Finally, the union over all classes of the polynomial-time hierarchy is denoted by PH. The polynomial-time hierarchy is itself contained in the class $\mathrm{P}^{\#\mathrm{P}}$ which we will discuss in the next section, and $\mathrm{P}^{\#\mathrm{P}}$, as well as all other decision problem complexity classes considered here, is contained in PSPACE, the class of problems that can be solved by a Turing machine with polynomially bounded space complexity. We have illustrated the structure of the complexity classes for decision problems considered here and later in Figure 2.4.

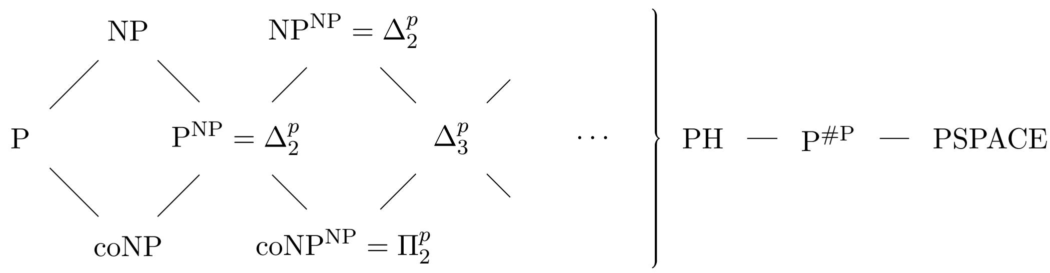


Figure 2.4: Illustration of the central decision complexity classes. Edges indicate inclusions from left to right.

### 2.1.2 Function and Counting Complexity

As mentioned at the beginning of the last section, a decision problem is the most restricted, non-trivial type of problem in terms of the number of possible answers. However, in many situations, we are interested in studying and solving problems of the form "*What is the value of $f(x)$?*" for a function $f$ and a given input $x$. This type of problems is called *function problems*. Here we are particularly interested in the special case of counting problems.

Suppose we are given a decision problem $A$. The counting variant of $A$, denoted by $\#A$, does not ask whether a solution exists, but how many solutions exist. Valiant [1979] introduced the central class of counting problems #P, which consists of the counting variants of decision problems in NP. For example, #SAT $\in$ #P, the counting variant of SAT, does not ask whether an assignment exists which sets the formula to true, but how many such assignments exist. The counterpart of P for function problems, that is the class of function problems which can be answered in polynomial time by a Turing machine, is denoted by FP. Even if we consider the classes FP and #P as counterparts of P and NP in the context of counting problems, this is of course not valid in general. In fact, FP is by definition not a subset of #P, for the simple reason that it also contains other function problems such as problems regarding calculating certain probabilities and also decision problems.

The standard reduction for function problems is the Turing reduction. A function problem $A$ is *polynomial-time Turing reducible* to a function problem $B$, denoted by $A \leq_T^p B$, if there exists a Turing machine which, with access to an oracle for $B$, can answer any instance of problem $A$ in polynomial time. Analogous to the previously defined hardness and completeness of decision problems with respect to the polynomial-time many-one reduction, we define the hardness and completeness of function problems with respect to complexity classes like #P using the polynomial-time Turing reduction. In addition, restricted polynomial-time Turing reductions are often considered. For counting problems in particular, the *polynomial-time parsimonious reduction* is often considered, which, similar to the polynomial-time many-one reduction, transforms an instance of the original problem into an instance of the target problem in polynomial time, such that the number of solutions in the constructed instance equals the number of solutions in the original instance.

It is evident that every decision problem in NP is polynomial-time Turing reducible to its counting variant in #P, since a Turing machine with access to the answer to the counting variant through the oracle can easily check whether a solution exists at all. Thus, it is clear that the counting variants of NP-complete decision problems are computationally hard since the existence of a polynomial-time algorithm for them would imply P = NP. However, what about the counting variants of decision problems that are in P? To answer this question, Valiant [1979] studied the complexity of the counting variant of the decision problem of checking whether a perfect matching, i.e., a subset of edges in which each node can be found exactly once, exists in a bipartite graph, that is, an undirected graph in which the nodes are divided into two disjoint sets and no edges exist between two nodes within the same set. Formally, the problem is defined as follows.

| Perfect-Bipartite-Matching | |
|---|---|
| **Given:** | A bipartite graph $G = (U, V, E)$ with $\lvert U\rvert = \lvert V\rvert = n$, and edges $E \subseteq U \times V$. |
| **Question:** | Does there exist a perfect matching $M \subseteq E$? |

While it is known that the decision variant of this problem is in P, Valiant [1979] showed that its counting variant #PERFECT-BIPARTITE-MATCHING is #P-complete with respect to the polynomial-time Turing reduction and thus probably not solvable in polynomial time. How the complexity of such #P-hard and #P-complete problems relates to the complexity of decision problems in the polynomial-time hierarchy was studied by Toda [1991]. He showed that an algorithm with a single call to a #P oracle can solve any problem from the polynomial-time hierarchy in polynomial time, whereby $\mathrm{PH} \subseteq \mathrm{P}^{\#\mathrm{P}}$ holds. Thus, the immense complexity of some function problems, such as the problems of computing certain probabilities, is often proved by showing their #P-hardness, even if they are not counting problems themselves and therefore are not in #P. Furthermore, it is also possible to show the #P-hardness of decision problems using the Turing reduction as we will see later.

### 2.1.3 Parameterized Complexity

The previous definitions of computational complexity may give the impression that there are essentially only two kinds of problems: simple ones that can be solved in polynomial time and computationally hard, e.g., NP-hard or #P-hard, problems that are so hopelessly complex that they cannot be solved for larger instances in practice. However, there are several approaches to further differentiate the complexity of problems, both theoretically and practically, and the most prominent approach for the worst-case complexity is to consider the *parameterized complexity* as formally introduced by Downey and Fellows [1992, 2012]. The key idea behind parameterized complexity is to understand how the complexity of problems depends on certain parameters of their inputs. The resulting insights often explain the efficient solvability of certain problems in practice, since parameters that make the problems difficult to solve in the worst-case may not occur in practice with the same magnitude or frequency.

In the following we will focus on the definitions concerning decision problems, the definitions for function problems and counting problems follow mostly analogously. A *parameterized (decision) problem* is a (decision) problem $A$ combined with a parameter $p$ that constrains the input. The parameter can be directly related to the input, like the desired size of the clique or the number of nodes in a given graph, or it can be an underlying parameter, like the treewidth of a given graph.

**Example 2.2.** For example, consider the *dominating set problem*, which is defined as follows.

| | DOMINATING SET |
|---|---|
| **Given:** | An undirected graph $G = (V, E)$ and $k \in \mathbb{N}$. |
| **Question:** | Does there exist a set $V' \subseteq V$ with $\lvert V'\rvert \leq k$ such that each $v \in V \setminus V'$ is adjacent to at least one $u \in V'$? |

Natural parameters which could be chosen to parameterize DOMINATING SET are the number of nodes $|V|$, the number of edges $|E|$, the maximum size of the dominating set $k$, the maximum node degree $\Delta$, or the treewidth of $G$.

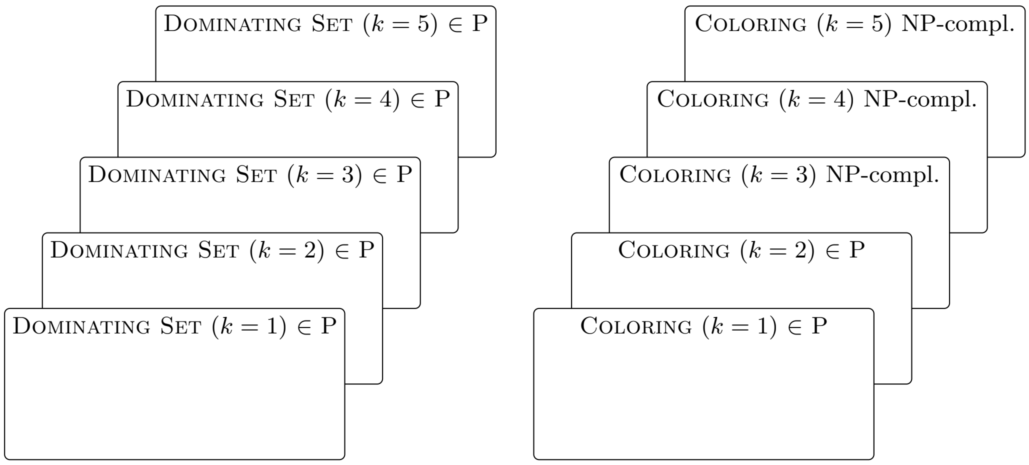


Figure 2.5: Complexity of the slices of Dominating Set and Coloring with respect to the parameter $k$, as discussed in Example 2.3.

The first approach we consider here is to study the complexity of parametrized decision problems in the case where the parameter is simply assumed to be a constant. The complexity class XP contains the parametrized decision problems that can be solved in $\mathcal{O}(g(p) \cdot |I|^{f(p)})$ for some computable functions $f$ and $g$, that is, polynomial time for a constant parameter $p$, using a deterministic Turing machine. A parameterized decision problem is referred to as *para-NP-hard* if and only if the decision problem is NP-hard for the case where the parameter is assumed to be constant. In the context of XP and para-NP-hardness, we often speak of examining the complexity in terms of so-called *slices* of the problem, where each slice of the problem corresponds to a particular value of the parameter.

**Example 2.3.** Consider Dominating Set parameterized by the desired maximum size $k$ specified in the input. As mentioned in the previous section, Dominating Set is NP-complete. However, we can try all $\sum_{i=1}^{k} \binom{n}{i} \leq (n+1)^k$ possible subsets $V' \subseteq V$ of size at most $k$ with $n = |V|$ and thus check whether such a dominating set exists and thus solve the problem. Assuming that $k$ is constant, this is possible in polynomial time, so it follows that dominating set parameterized by $k$ is in XP.

On the other hand, consider Coloring, the problem of checking whether a coloring of the nodes of a graph with a given number of colors $k$ exists such that no adjacent nodes share the same color. Lovasz [1973] showed that the problem is NP-hard for fixed $k \geq 3$, implying that Coloring parametrized by the number of colors is actually para-NP-hard, and thus presumably not in XP.

We have summarized the complexity of the first few slices of the two problems with respect to parameter $k$ in Figure 2.5.

While the distinction between membership in XP and para-NP-hardness already provides some insight into the complexity of NP-hard problems, there exists a further popular classification for problems in XP. This is reasonable since even if a problem is in XP, its complexity can scale very poorly with respect to the input size due to the potentially parameter-dependent degree of the polynomial constraining the runtime of a respective algorithm, such as in the case of the brute force algorithm in Example 2.3 for Dominating Set.

We say that a parameterized (decision) problem with parameter $p$ is *fixed parameter tractable* (FPT) if it can be solved in $\mathcal{O}(f(p) \cdot |I|^{\mathcal{O}(1)})$ for some computable function $f$, often referred to as *FPT-time*, using a deterministic Turing machine. The complexity class of decision problems that are fixed parameter tractable is denoted by FPT accordingly. Thus, compared to the surrounding class XP, the portion of the runtime that grows with the parameter is required to be limited to an isolated factor detached from the portion that is bounded by a fixed polynomial depending only on the input size.

**Example 2.4.** Consider the *vertex cover problem* defined as follows.

| Vertex Cover | |
|---|---|
| **Given:** | A graph $G = (V, E)$ and $k \in \mathbb{N}$. |
| **Question:** | Does there exist a set $V' \subseteq V$ with $|V'| \leq k$ such that for each $\{u, v\} \in E$ : $u \in V' \vee v \in V'$? |

As in the previous example for dominating set, one could check all $\sum_{i=1}^{k} \binom{n}{i} \leq (n+1)^k$ possible subsets $V' \subseteq V$ of size at most $k$ with $n = |V|$ to check the existence of a vertex cover, whereby Vertex Cover parameterized by $k$ is in XP. On the other hand, this algorithm does not run in FPT-time.

However, there is an alternative approach for Vertex Cover. Since for each edge $\{u, v\} \in E$ either $u$ or $v$ (or both) must be in $V'$ we can employ the following approach: As long as $k > 0$ and there is still an edge, choose an arbitrary edge $\{u, v\} \in E$, and repeat this for $k \leftarrow k - 1$ once in $G$ without $u$ and adjacency edges and once in $G$ without $v$ and adjacency edges. If we have removed all edges from the graph in one of the possible paths, we have found a vertex cover of size at most $k$, if not, such a vertex cover does not exist.

Each instance of the algorithm recursively invokes two new instances, giving us a total of $\sum_{i=0}^{k} 2^i = 2^{k+1} - 1$ instances, and in each instance of the algorithm, we only need $\mathcal{O}(n)$ additional steps, resulting in a total runtime of $\mathcal{O}(2^k \cdot n)$, which is FPT-time with respect to parameter $k$. Thus, Vertex Cover parameterized by $k$ is in FPT.

The corresponding reduction concept is the FPT-reduction. We say that a parameterized decision problem $A$ with parameter $p_A$ is *FPT-reducible* ($\leq_{\text{FPT}}$) to a parameterized decision problem $B$ with parameter $p_B$ if there exists a Turing machine which transforms every instance $I$ of $A$ with parameter $p_A$ into an instance $I'$ of $B$ with parameter $p_B \leq h(p_A)$ for some computable function $h$ in FPT-time with respect to $p_A$, where $I$ is a yes-instance of $A$ if and only if $I'$ is a yes-instance of $B$.

As for P and NP-hardness and for XP and para-NP-hardness, there is also a hardness notion which is opposed to a membership in FPT. For this Downey and Fellows [1992, 2012] introduced the W *hierarchy*. To define it, we first consider the following decision problem.

| Weighted Circuit Satisfiability | |
|---|---|
| **Given:** | A Boolean circuit $C$ over a set of Boolean variables $X$ and $k \in \mathbb{N}$. |
| **Question:** | Does there exist an assignment of $X$ with $k$ variables set to true, such that $C$ is satisfied? |

For a given Boolean circuit, we denote as the *depth* the maximum length of a path from an input to the output gate and as the *weft* the maximum number of gates on the path from an input to the output gate with in-degree of at least three. Each class of the W hierarchy W[$t$] with $t \geq 1$ consists of those parameterized decision problems that are FPT-reducible to the weighted circuit satisfiability problem parameterized by $k$ restricted to Boolean circuits with a constant depth $d \geq 1$ and weft $t$. We refer to a parametrized decision problem as W[$t$]-hard if all problems in W[$t$] can be reduced to the problem via FPT-reductions. Accordingly, a parameterized decision problem is W[$t$]-complete if the problem is W[$t$]-hard and is itself a member of W[$t$]. As before for NP, W[$t$]-hardness and membership in W[$t$] can be shown using reductions from problems already proven to be W[$t$]-hard or to problems already proven to be in W[$t$], respectively, due to the transitivity of the FPT-reduction. Prominent examples are Clique parameterized by the clique size which is W[1]-complete and Dominating Set parameterized by the maximum size of the set which is W[2]-complete. We illustrate the concept of reducing parametrized problems to Boolean circuits presented here using Dominating Set in Figure 2.6.

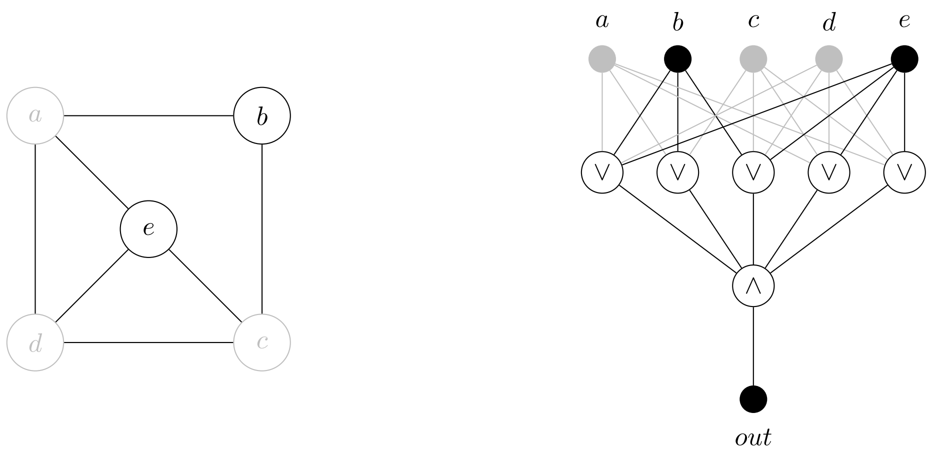


Figure 2.6: On the left we have a graph $G = (V, E)$ over $V = \{a, b, c, d, e\}$ together with a dominating set $V' = \{b, e\}$ of size $k = 2$. On the right side we have a corresponding Boolean circuit with a depth of 3 and a weft of 2. The satisfying assignment of the Boolean variables corresponding to the dominating set with $k = 2$ variables set to true is highlighted.

It holds that FPT $\subseteq$ W[1] $\subseteq$ W[2] $\subseteq \cdots \subseteq$ XP. While it is believed that each level of the W hierarchy including FPT and XP is a proper subset of the next level, only FPT $\subset$ XP has been proven so far. However, under this assumption, the W[$t$]-hardness of a parametrized decision problem is assumed to contradict the existence of a respective FPT-algorithm. Thus, we have seen that the four basic NP-complete problems we have considered here as examples, namely Coloring, Vertex Cover, Clique and Dominating Set seem to have quite different complexities with respect to their natural parameter $k$.

### 2.1.4 Average-Case Complexity

While computational complexity may seem like a self-justifying theory of reductions, complexity classes, and various abstract concepts that exists only for itself, one of its original goals was, and still is, to examine how long it takes to solve problems *in practice*. However, the relevance of the definitions presented here so far is questionable in this regard since they examine only the worst-case complexity of problems. While this approach is reassuring for problems that can be solved very efficiently even in the worst-case, it is very unsatisfying for problems that are NP-hard or #P-hard as it is often observed that real-world instances of such problems can be solved much faster than their worst-case computational hardness suggests. In fact, we will observe this ourselves in Chapter 6 for NP-hard and #P-hard problems in the context of tournaments using experiments on real-world data. Thus, to find a theoretical explanation for these observations Levin [1986] introduced the *average-case complexity*.

In the following, we will define the aspects of Levin's average-case complexity that are relevant to us in this work. We will use the conventions commonly used nowadays, such as those suggested by Impagliazzo [1995]. In particular, we have been guided here by the survey by Bogdanov and Trevisan [2006] and the chapter in the book by Arora and Barak [2009].

A *distributional (decision) problem* is given by a (decision) problem combined with a probability distribution $\mathcal{D}$ which models the distribution of the instances. It cannot be stressed enough that both the problem and the distribution are essential for the significance and complexity of distributional problems. For example, the distribution may be collapsed to one possible instance, produce only trivial instances, or have other nonsensical properties with respect to the actual objectives of the investigation or the scenario.

In the original definition by Levin [1986], $\mathcal{D}$ is given by single distribution function $\mu : \mathbb{N} \to [0, 1]$, along with the respective density function $\mu'(x) = \mu(x) - \mu(x - 1)$, over a numeration over all instances of variable size. For convenience, we simply denote the probability of an instance $I_x$ by $\mu'(I_x) = \mu'(x)$. However, while this definition allows to model that different input sizes occur with different probabilities, the definition is sometimes inconvenient from a conceptual point of view for analyses similar to the previous approaches for worst-case complexity, for which the instances are usually segmented with respect to their sizes. Thus, Impagliazzo [1995] proposed that instead of a single distribution, $\mathcal{D}$ is given by a sequence of distributions $\{\mathcal{D}_n\}_{n\in\mathbb{N}}$, a so-called *ensemble*, where $\mathcal{D}_n$ denotes the distribution for instances of size $n$. Fortunately, he also demonstrated that this convention makes no essential difference for most of Levin's definitions.

Something that Levin left open in his original definition, but which is of essential theoretical relevance, is the complexity of $\mathcal{D}$ itself. Ben-David et al. [1992] defined two different types of distributions for this purpose. A distribution $\mathcal{D}$ is *P-computable* if for a given instance the distribution function, and hence the density function, can be computed in polynomial time with respect to the instance size. However, there are many prominent distributions for which calculating those probabilities is #P-hard. One example for this is the uniform distribution over the completion of partial orders (see Brightwell and Winkler [1991]). Thus, they have defined yet another broader category of distributions. A distribution $\mathcal{D}$ is *P-samplable* if there exists an algorithm which, for a given input size $n$, generates a random instance of size $n$ with respect to $\mathcal{D}$ in polynomial time with respect to $n$.

Now we have everything at hand to define the average-case complexity of algorithms and problems. We say that an algorithm $A$ runs in *expected polynomial time* with respect to a P-samplable distribution $\mathcal{D}$ if $\mathrm{E}_{\mathcal{D}_n}[T_A] = \sum_{|I|=n} \mu'_{\mathcal{D}_n}(I) \cdot T_A(I) \in \mathcal{O}(n^k)$ for some fixed $k \in \mathbb{N}$ where $\mathrm{E}_{\mathcal{D}_n}[T_A]$ denotes the expected value of the running time $T_A$ of $A$ with respect to the distribution $\mathcal{D}_n$. Likewise, we say that a distributional problem can be solved in expected polynomial time if a corresponding algorithm exists.

While this definition over the pure expected running time is intuitive and often appropriate, it poses a problem when trying to define complexity classes as in the previous sections. For example, algorithms that have an expected polynomial time for a distribution may suddenly have an exponentially growing expected running time after a quadratic slowdown, whereby the definition is not robust with respect to polynomial changes of the running time. This is problematic for the definition of complexity classes insofar as the classification of the running time depends very strongly on the underlying computational model and polynomial-time reductions of problems to problems with polynomial expected running time can result in exponential running times.

Thus, Levin introduced the following, slightly more unintuitive but robust, definition. We say that an algorithm $A$ runs in *average-case polynomial time* with respect to a P-samplable distribution $\mathcal{D}$ if $\mathrm{E}_{\mathcal{D}_n}[(T_A)^{\varepsilon}] = \sum_{|I|=n} \mu'_{\mathcal{D}_n}(I) \cdot (T_A(I))^{\varepsilon} \in \mathcal{O}(n)$ for some fixed $\varepsilon > 0$. Accordingly, we say that a distributional problem can be solved in average-case polynomial time if a corresponding algorithm exists. The class distNP is defined as the class of distributional decision problems consisting of a problem from NP combined with a P-samplable distribution. The class AvgP (or sometimes distP) is defined as the class of distributional decision problems in distNP that can be solved in average-case polynomial time. Note that any algorithm that runs in expected polynomial time also runs in average-case polynomial time.

As in the previous sections, there is a corresponding type of reduction for AvgP and distNP. It is defined analogously to the polynomial-time many-one reduction with the addition of the so-called domination condition with respect to the two probability distributions, which ensures that instances with high probabilities are not mapped to instances with too low probabilities. If this condition were missing, it would be possible to lower the complexity of the problem in the reduction simply by drastically reducing the probabilities of certain instances. Thus, this condition is required to ensure that a distributional problem that is reduced to a distributional problem in AvgP is also contained in AvgP. Using this reduction, it is then possible to define distNP-hardness and distNP-completeness. However, we will not further elaborate on this here, since we are mainly focused on finding expected polynomial time and average-case polynomial time algorithms for worst-case computational hard problems.

As mentioned previously, the complexity of distributional problems depends crucially on the distribution chosen. In practice, however, it is often difficult to find a meaningful distribution that approximates the real-world distribution of instances or at least dominates it in terms of complexity. A promising approach in this case is the smoothed complexity analysis as introduced by Spielman and Teng [2004, 2009]. We will discuss possible models and applications of this approach in the area of computational social choice in Chapter 7.

## 2.2 Computational Social Choice

We will now introduce the basics for elections and tournaments from the field of computational social choice, on which we will later build our investigations. As mentioned in the introduction, one of the key aspects of computational social choice is the study of problems related to decision making processes from a computational point of view. While elections play a very prominent role, they are by far not the only topic considered. Other extensively studied topics include tournaments, resource allocation, matching under preferences, hedonic games, and judgment aggregation. As previously discussed, the developments in recent decades such as the digitization of elections, the introduction of direct democratic participation processes, and the establishment of social networks have resulted in unprecedented opportunities and risks along with immense amounts of data, making the computational aspects of the respective processes more important than ever before. Naturally, we can only address a small fraction of the numerous topics and literature in the field of computational social choice. For a comprehensive overview, we refer to the *Handbook of Computational Social Choice* edited by Brandt et al. [2016], *Economics and Computation* edited by Rothe [2016], and *Trends in Computational Social Choice* edited by Endriss [2017].

### 2.2.1 Elections

An *election* is given by a tuple $E = (C, V)$ with the set of *candidates* $C = \{c_1, \ldots, c_m\}$ with $m \geq 2$ and a list of votes, also called *preference profile*, $V = (v_1, \ldots, v_n)$ with $n \geq 1$, where $v_i$ denotes the *vote* (or *preference*) of *voter* $i \in N$ with $N = \{1, \ldots, n\}$. Throughout this work, if not stated otherwise, $m$ and $n$ are used by default to denote the number of candidates and the number of voters/votes, respectively. We focus on the two most prominent types of votes used in the literature, namely approval votes and ordinal votes. In the case of *approval votes*, each vote is given by a vector $v_i \in \{0, 1\}^m$, where the $j$-th entry indicates the disapproval or the approval of voter $i$ for candidate $c_j$ with a 0 or 1, respectively. In the case of *ordinal votes*, each vote $v_i$ is given by a complete linear order $>_i$ over the set of candidates $C$. Thus, each voter specifies a complete ranking over the candidates from most preferred at position one down to least preferred at position $m$.

In the following we present an example for an election using ordinal votes.

**Example 2.5.** Consider the election $E = (C, V)$ with the set of candidates $C = \{a, b, c, d\}$ and preference profile $V = (v_1, v_2, v_3, v_4, v_5)$. The ordinal votes are given as follows.

$$\begin{aligned} v_1 &: \ b > a > d > c \\ v_2 &: \ c > a > b > d \\ v_3 &: \ c > a > b > d \\ v_4 &: \ d > c > b > a \\ v_5 &: \ a > d > b > c \end{aligned}$$

The vote $v_3 : c > a > b > d$ of voter 3 indicates that voter 3 reported candidate $c$ at position 1 as the most preferred candidate, followed by candidate $a$ at position 2, candidate $b$ at position 3, and candidate $d$ at position 4 as the least preferred candidate.

For the analysis of elections from a computational point of view, the encoding of an election is of immense importance. As common in the literature, we assume by default that the set of candidates, the profile, and the votes are given as lists and thus an election can be encoded for both approval and ordinal votes in size $\mathcal{O}(n \cdot m)$. Alternatively, the profiles can be encoded in the *succinct representation*, where the different available votes together with their multiplicities as binary numbers are given. Thus, the input size can be exponentially smaller than in the list representation. Hardness results are transferred from the standard representation to the succinct representation, whereas efficiency results are transferred from the succinct representation to the standard representation. However, the actual complexity of problems can differ for the two representations, see Fitzsimmons and Hemaspaandra [2017].

In some cases, we will also encounter *partial profiles*, which, unlike the complete profiles defined previously, may contain partial votes. A *partial approval vote* is given by a vector $\tilde{v}_i \in \{0, 1, \perp\}^m$ where $\perp$ indicates an undetermined or missing entry. A *partial ordinal vote* is given by a partial order $\tilde{v}_i$ over the candidates $C$, which is a irreflexive and transitive, but, in contrast to a linear order, not necessarily connex relation. Thus, it is possible that for certain pairs of candidates the preference of the respective voter is missing. For both types, a *completion* of a partial vote $\tilde{v}_i$ is a complete vote $v_i$ of the respective type in which the missing entries/comparisons have been determined while the previously determined ones have been carried over. A completion of a partial profile $\tilde{V} = (\tilde{v}_1, \dots, \tilde{v}_n)$ is a complete profile $V = (v_1, \dots, v_n)$ in which $v_i$ is a completion of $\tilde{v}_i$ for $1 \leq i \leq n$.

The winners of a given election are determined using a voting rule. Formally, a *voting rule* is given by a mapping $\mathcal{E} : \mathcal{V}(C)^n \to 2^C \setminus \{\emptyset\}$, which maps a given list of votes over $C$ to a non-empty set of winning candidates, where $\mathcal{V}(C)$ denotes the set of all possible votes over $C$ for the respective vote type. Note, however, that we focus on single-winner elections and thus the candidates in the set of winners should not be interpreted as being on a committee but as being in a tie. In the literature, and also in the following, the term voting rule usually not only refers to a single voting rule for a specific combination of $m$ and $n$, but to an entire family for multiple different combinations. Here, we focus on the following voting rules.

In the case of approval votes, we consider the canonical *approval voting* (AV) rule, where the candidates with the maximum number of approvals win. While AV satisfies many desired theoretical and practical properties, it is rarely used in practice (see Zwicker [2016]). In addition, we consider a variant of AV, referred to as *$k$-AV* with $m > k$, in which each voter must allocate exactly $k$ approvals. Of special interest here are the most prominent single-winner voting rules, namely *plurality*, which corresponds to 1-AV, and *veto*, which corresponds to $(m-1)$-AV.

In the case of ordinal votes, we focus on a family of voting rules referred to as positional scoring rules. A *positional scoring rule*, or simply *scoring rule* for short, for a given number of candidates $m$, is defined by a *scoring vector* $\vec{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_m) \in \mathbb{N}_0^m$ with $\alpha_1 \geq \alpha_2 \geq \dots \geq \alpha_m$ with $\alpha_1 > \alpha_m$. The so-called *score value* $\alpha_j$ denotes the amount of points a candidate receives for being placed in position $j$ by a voter. By accumulating these points, we obtain the *score* of the candidates given by $\text{score}(c) = \sum_{i=1}^{n} \alpha_{\text{pos}_i(c)}$, where $\text{pos}_i(c)$ denotes the position of candidate $c$ in vote $v_i$. The winners are the candidates with the maximum score. A scoring rule for a variable number of candidates is characterized by a function that generates the respective scoring vector for the given number of candidates in polynomial time. A class of scoring rules, which includes all natural and common scoring rules, are the *pure* *scoring rules* introduced by Betzler and Dorn [2010]. The restriction here is that starting from an initial vector for the minimum number of candidates $m_0$ for which the rule is defined, the respective vector for $m+1$ candidates has to be generated by inserting a valid value into the vector for $m$ candidates. The most prominent (pure) scoring rules are the following.

*$k$-approval.* Each voter awards one point to each of the first $k \geq 1$ candidates in his or her preference, whereby $\vec{\alpha} = (\underbrace{1, 1, \ldots, 1}_{k\text{-times}}, \underbrace{0, 0, \ldots, 0}_{(m-k)\text{-times}})$ for $m > k$.

The well-known *plurality* rule corresponds to 1-approval.

*$k$-veto.* Each voter awards one point to each candidate except for the last $k \geq 1$ in his or her preference, whereby $\vec{\alpha} = (\underbrace{1, 1, \ldots, 1}_{(m-k)\text{-times}}, \underbrace{0, 0, \ldots, 0}_{k\text{-times}})$ for $m > k$.

The well-known *veto* rule corresponds to 1-veto.

*Borda.* Each voter allocates the points to the candidates according to the number of lower-ranked candidates, whereby $\vec{\alpha} = (m-1, m-2, \ldots, 0)$.

*$(2, 1, \ldots, 1, 0)$.* Each voter allocates one approval to the first candidate and one disapproval to the last candidate in his or her preference, whereby $\vec{\alpha} = (2, 1, \ldots, 1, 0)$.

We illustrate these definitions in the following example.

**Example 2.6.** Consider the election $E = (C, V)$ given in Example 2.5. If we apply Borda with scoring vector $\vec{\alpha} = (3, 2, 1, 0)$ as the voting rule to $E$ we receive the following scores.

$$\begin{aligned}
\text{score}(a) &= 2+2+2+0+3 = 9 \\
\text{score}(b) &= 3+1+1+1+1 = 7 \\
\text{score}(c) &= 0+3+3+2+0 = 8 \\
\text{score}(d) &= 1+0+0+3+2 = 6
\end{aligned}$$

Thus, we see that $a$ is the unique winner of election $E$ with respect to Borda in this case.

Note that if we had applied $(2, 1, \ldots, 1, 0)$ to the profile in Example 2.5 instead of Borda, candidate $b$ would have been the unique winner of the election. Thus, the outcome of an election can strongly depends on the voting rule chosen and, here in particular, the scoring rule chosen. We will discuss this issue in much more detail in Chapter 4.

If we look closely at the scoring rules, we notice that a rule such as plurality is characterized not only by the vector $(1, 0, \ldots, 0)$ but also by $(2, 0, \ldots, 0)$, just as Borda is characterized not only by $(m-1, m-2, \ldots, 1, 0)$ but also by the, occasionally used, vector $(m, m-1, \ldots, 2, 1)$. The reason for this is that two scoring rules $\vec{\alpha}, \vec{\alpha}' \in \mathbb{N}_0^m$ are equivalent with respect to their winner determination if but also only if they can be mapped to each other using the transformation $\vec{\alpha}' = a \cdot \vec{\alpha} + b$ with $a \in \mathbb{Q}_{>0}$ and $b \in \mathbb{Z}$. Now if we say that a certain property or result holds true for a certain scoring rule, then it is meant that it holds true for this rule and all equivalent ones. Based on the previous equivalence, one can also normalize all equivalent scoring rules to a unique representative with $\alpha_m = 0$ and $\alpha_1, \ldots, \alpha_{m-1}$ scaled such that their greatest common divisor equals one.

If we consider the approval vote voting rule $k$-AV and the ordinal vote voting rule $k$-approval for a given $k$ directly side by side, it becomes clear that both describe essentially the same voting rule. The crucial difference between the two is only the type of the votes and the different strength of expression. In fact, the ordinal vote in this case is clearly more expressive than the respective approval vote, which can be seen from the fact that the former can easily be reduced to the latter, but the reverse does not apply. We will see later in Chapter 5 that this very difference can have massive effects on the respective complexity of voting problems.

As mentioned in the definition of voting rules, we allow the rules to output a non-empty set of tied winners. Indeed, it is easy to show that under the assumption of basic fairness criteria such as equal treatment of voters and candidates, no voting rule can be resolute, i.e., always output exactly one winner. However, since the ultimate goal of single-winner elections is to determine a single winner, we must eventually break these ties using *tie-breaking*. A tempting approach to this is to further refine the winner set by applying further comparison criteria. For example, one could try to break the ties of a plurality election by comparing the Borda scores of the winners. However, such approaches can also be expressed as single voting rules, which brings us back to the theoretical limit mentioned above. Leaving fairness aside, a frequently used approach in both theory and practice is lexicographic tie-breaking, i.e. breaking ties based on a given order. Another frequently used approach to break ties in a fair manner, despite the theoretical limits mentioned above, is random tie-breaking. Namely, we conduct a lottery among the tied candidates to determine the winner. Here, one can either give the candidates uniform probabilities or, more generally, use certain criteria to determine those probabilities. For an overview and discussion on tie-breaking approaches we refer to the paper by Freeman et al. [2015]. In addition, we propose the use of robustness metrics, which we will consider in Chapters 3, 4, and 5, as a possible very meaningful comparison criterion to break ties or to determine the probabilities for the tie-breaking lottery.

Note that this section is only a brief glimpse into the rich area of voting. For a comprehensive overview, we refer to the chapter by Zwicker [2016].

### 2.2.2 Computational Problems

At the heart of computational social choice lies the study of formal election problems from the perspective of computational complexity. In the following, we will introduce and discuss the central problems.

#### Winner Determination

When studying elections from a computational point of view, probably the most striking problem to study is the problem of determining the winners of an election with respect to a certain voting rule. Consequently, the initial investigation of this problem by Bartholdi, Tovey, and Trick [1989b], along with their work on manipulation which we will discuss afterwards, laid the foundation of computational social choice. For a voting rule $\mathcal{E}$, the *winner determination problem* is given as follows.

| $\mathcal{E}$-Winner | |
|---|---|
| **Given:** | An election $E = (C, V)$ and a distinguished candidate $p \in C$. |
| **Question:** | Is $p$ a winner of the election $E = (C, V)$ with respect to $\mathcal{E}$? |

For voting rules such as the approval voting rule, the family of scoring rules, and many other well-known voting rules such as Copeland, the winner determination problem is decidable in polynomial time, usually by direct evaluation of the definitions and formulas with respect to the given election. However, Bartholdi et al. [1989b] showed that the winner determination problem for the Dodgson voting rule, introduced by Charles Dodgson [1876], better known as Lewis Carroll, and for the Kemeny voting rule, introduced by Kemeny [1959], is indeed NP-hard. Later, Hemaspaandra et al. [1997] and Hemaspaandra et al. [2005] showed that the winner determination problems for Dodgson and Kemeny are not only NP-hard, but in fact $\Theta_2^p$-complete, whereas $\Theta_2^p$ denotes the class of decision problems that can be solved in polynomial time by a deterministic Turing machine with parallel access to an NP oracle. An interesting case is the single transferable vote voting rule (STV), for which the winner determination problem is NP-complete according to Conitzer et al. [2009], but a single winner can be determined in polynomial time. The reason for this is that during the evaluation of STV one may have to break multiple ties, which can result in exponentially many different paths with possibly different winners at the end. Following an arbitrary path to determine a single winner is possible in polynomial time. However, it seems that checking whether a particular candidate ends up as the winner in one of the potentially exponentially many paths is computationally hard.

### Manipulation

In parallel to the winner determination problem, Bartholdi et al. [1989a] also introduced the first election interference problem, the manipulation problem. For a voting rule $\mathcal{E}$, the *manipulation problem* is given as follows.

| $\mathcal{E}$-Manipulation | |
|---|---|
| **Given:** | An election $E = (C, V)$ and a distinguished candidate $p \in C$. |
| **Question:** | Is there a vote $v \in \mathcal{V}(C)$, such that $p$ is a winner of the election $E' = (C, V')$ with $V' = V \cup (v)$ with respect to $\mathcal{E}$? |

As mentioned in the introduction, the main motivation for the complexity-theoretic investigation of the manipulation problem were the results of Gibbard [1973] and Satterthwaite [1975] and, later, Duggan and Schwartz [2000], which, loosely speaking, state that any somewhat appropriate voting rule is susceptible to manipulation. Bartholdi et al. [1989a] therefore suggested a possible computational hardness as a possible barrier against manipulation. They showed that the problem is decidable in polynomial time for scoring rules and Copeland, among others, using a greedy algorithm, while the problem is NP-complete for the second-order Copeland rule. This is in so far interesting, as the winner determination problem for second-order Copeland is decidable in polynomial time and thus the hardness does not arise from the implicit solution of that very problem through the manipulation problem, but from the problem of manipulation itself. The extension of the manipulation problem to multiple cooperating manipulators is referred to as Coalitional Manipulation and was introduced by Conitzer et al. [2007]. For a comprehensive overview on the manipulation problem and the coalitional manipulation problem, especially regarding computational results, we refer to Conitzer and Walsh [2016]. Note that the problem of manipulation is not malicious per se. If we want the votes of voters to correspond as closely as possible to their underlying true preferences, so that we can achieve a satisfactory collective result in this respect, then manipulation by individual voters is obviously negative and a computational barrier is desirable. However, in practice, agents rarely vote this way and mostly try to maximize their personal utility. Thus, it can be argued that it is up to each voter to understand the impact of his or her vote and thereby, from a fairness and transparency perspective, a computational hurdle is not desirable. In this respect, the less condemning term *strategic voting* is often used.

Regarding election interference problems such as manipulation, there is usually the so-called *constructive variant*, where we try to make a distinguished candidate the winner, as in the definition of the manipulation problem presented above, and the *destructive variant*, where we try to prevent a distinguished candidate from winning. Further, there is a distinction as to when the distinguished candidate is a winner, which we refer to as the *winner model*. In the *unique winner case*, the interference is considered successful if, in the constructive case $\{p\} = \mathcal{E}(V')$ and in the destructive case $\{p\} \neq \mathcal{E}(V')$. On the other hand, in the *non-unique winner case*, the interference is considered successful if, in the constructive case $p \in \mathcal{E}(V')$ and in the destructive case $p \notin \mathcal{E}(V')$. Thus, in the constructive unique winner case and in the destructive non-unique winner case we are pessimistic about the tie-breaking and want to guarantee that the distinguished candidate wins or respectively cannot become the winner after the tie-breaking. In the constructive non-unique winner case and in the destructive unique winner case we are optimistic about the tie-breaking and hope that the distinguished candidate will win or respectively will not win after the tie-breaking. Typically, results focus on one of the two models, but they can often be transferred to the other case as well.

### Control

Following manipulation, Bartholdi et al. [1992] also introduced the interference problem of *election control*. Here, the election chair tries to influence the outcome of the election by deliberately changing the election process. They considered constructive election control by adding or deleting candidates or voters and partitioning the candidates or voters creating two primary elections. Since the different classical variants of election control are only of limited relevance for the results here, we refer to the chapter by Faliszewski and Rothe [2016] for a comprehensive overview. Another type of election control by the chair is the intentional design of voting rules which we will consider later in Chapter 4.

### Bribery

The third central election interference problem is *bribery*, which was introduced more recently by Faliszewski et al. [2006] and is also studied by us in this work. Therefore, we introduce and discuss it in much more detail in the following Section 2.2.3.

### Possible Winner

Turning away from the interference problems, another central problem in computational social choice and relevant to many of our results is the possible winner problem, which was introduced by Konczak and Lang [2005]. For a voting rule $\mathcal{E}$, the *possible winner problem* is defined as follows.

| $\mathcal{E}$-Possible Winner | |
|---|---|
| **Given:** | A set of candidates $C$, a partial profile $\tilde{V}$ over $C$, and a distinguished candidate $p \in C$. |
| **Question:** | Is there a completion $V$ of $\tilde{V}$, such that $p$ is a winner of the election $E = (C, V)$ with respect to $\mathcal{E}$? |

The scenarios in which we need to perform a possible winner determination on partial data are manifold. They range from election predictions based on incomplete voter preferences aggregated from social media to winner determination based on incomplete election data due to social and political circumstances in the area or flaws in data collection or transmission. The possible winner problem concerning (pure) scoring rules is actually very well studied in terms of its complexity. Betzler and Dorn [2010] showed that the problem is NP-complete for all pure scoring rules except plurality, veto and $(2, 1, \dots, 1, 0)$ whereas it is in P for plurality and veto. Baumeister and Rothe [2012] completed the dichotomy result by showing that the problem is also NP-complete for $(2, 1, \dots, 1, 0)$. Recently, Chakraborty and Kolaitis [2021] have strengthened the dichotomy result by showing that it also holds if one assumes that the partial votes of the voters are partial chains, i.e., each vote consists of a complete order over a subset of the candidates.

For AV and $k$-AV, on the other hand, the possible winner problem is in P. In the case of AV, one only has to check whether the distinguished candidate is a winner if one sets all undetermined entries for it to 1 and for all other candidates to 0. For $k$-AV, the problem is a bit more complicated, since one also has to ensure that each vote contains exactly $k$ approvals after completion. First, we set each undetermined entry for the distinguished candidate for which it is possible to 1, and in the next step we use a straightforward flow network to check if it is possible to distribute the remaining approvals among the undetermined entries such that no other candidate has more (or at least as many) approvals as the distinguished candidate.

In addition, many special cases and variants of the possible winner problem have been studied in the literature. For example, Chevaleyre et al. [2010] studied the complexity of the problem under the assumption that new candidates are added, thus each voter has a complete order over the existing candidates but the preferences regarding the new ones are missing. The difference to the special case of Chakraborty and Kolaitis [2021] discussed above is that here the subset of candidates considered is the same for all voters. Kalech et al. [2011] studied the special case where voters indicate only their top-$k$ candidates. Moreover, the problem was also applied to cases other than partial votes. For example, Baumeister et al. [2012] and recently Neveling et al. [2021] studied the problem for uncertain weights of voters in weighted elections, and Baumeister et al. [2011b] studied the problem for uncertainty about the voting rule. The latter variant is also studied by us in Chapter 4.

However, the possible winner problem as previously defined is only of limited use if a justified winner actually has to be determined using partial data. For example, it may be that several candidates or in the worst case all the candidates are possible winners even though preferences are available. We present a possible solution to this problem in Chapter 5, where we not only examine whether the candidates are possible winners, but also how likely they are to win under certain assumptions, making them comparable.

For a general overview regarding the possible winner problem in elections, we refer to the recent survey by Lang [2020]. The possible winner problem has also been considered in other areas than elections, most notably in sports, where it is referred to as the elimination problem. We will introduce this problem later in Section 2.3 and present our results on it in Chapter 6.

### 2.2.3 Bribery

While the previously presented concept of manipulation is the most studied problem of election interference, especially in terms of its axiomatic and computational aspects, it makes a very strong assumption: the manipulator is involved so deeply in the election or in the elicitation of votes that he or she can specify a certain set of votes however he or she wants. The question is whether there is a more subtle way for an agent to influence the election. Perhaps, as a first step, the agent could kindly ask some voters to change their votes, but since this would probably not be very successful, it seems reasonable to offer the voters a reward for their willingness to change their votes, thus a trade, which leads us to the concept of bribery. Note, however, that 'the trade' does not have to be the naïve act of handing over a certain amount of currency in exchange for the service, whereby the boundaries of *malicious* bribery quickly become blurred. For example, the offering provided by the agent can also be a simple giveaway during a promotion event, which may subliminally influence the voter's preference, a campaign by a party in order to increase its popularity or decrease the popularity of the opponents, or the mere gift, or at least the illusion, of personal attention. Bribery in its various forms was already part of everyday political life in ancient Rome (Lintott [1990]) but is also not unknown in today's democracies (Lehoucq [2003]). Especially today's possibility to reach a large number of people quickly and anonymously via the Internet provides a great environment for the various forms of bribery.

The formal definition of bribery considered below in the context of computational social choice is derived from the original definitions by Faliszewski et al. [2006, 2009a] and the subsequent work by Faliszewski [2008], Elkind et al. [2009], and Faliszewski et al. [2009b]. The definition considered here, based on the survey by Faliszewski and Rothe [2016], generalizes the previous definitions just mentioned in that it provides for a individual generic price function for each voter. A *price function* for a certain voter is given by a mapping $\Pi : \mathcal{V}(C) \to \mathbb{N}_0$ with $\Pi(v) = 0$ for the initial preference $v$ of the voter and $\mathcal{V}(C)$ denoting the set of possible preferences over $C$ with respect to the given preference type. In that sense, a price function specifies how much the respective voter demands to change his or her initial preference $v$ into $v'$. Given a voting rule $\mathcal{E}$, the *priced bribery problem* is defined as follows.

| $\mathcal{E}$-Priced-Bribery | |
|---|---|
| **Given:** | An election $E = (C, V)$ with profile $V = (v_1, \dots, v_n)$, a distinguished candidate $p \in C$, a budget $B \in \mathbb{N}_0$, and a collection of price functions $\Pi = (\Pi_1, \dots, \Pi_n)$. |
| **Question:** | Is there a profile $V' = (v'_1, \dots, v'_n)$ over $C$ with $\sum_{i=1}^{n} \Pi_i(v'_i) \leq B$, such that $p$ is a winner of the election $E = (C, V')$ with respect to $\mathcal{E}$? |

Note that this definition must be seen as a generic framework that is then tailored to the specific bribery problem, especially with respect to the preference type, the unique winner or non-unique winner case, the price function itself, and especially the encoding of the price function. In some cases, additional restrictions may apply in addition to the budget. Moreover, the destructive variant is also considered, in which we have to ensure that $p$ will not be a winner of the election with respect to the winner case after the interference. To bring this definition to life, we will now discuss the bribery models in the previously mentioned work, which have led to this more general definition.

As previously mentioned, Faliszewski et al. [2006, 2009a] initiated the study of the computational complexity of bribery in elections. They consider both approval as well as ordinal elections. For their definition, referred to as Bribery, the discrete price function is used with $\Pi_i(v'_i) = 0$ if $v_i = v'_i$ and 1 otherwise, which simply limits the total number of voters for which we can change the vote by $B$. Moreover, they also considered \$Bribery with price function $\Pi_i(v'_i) = 0$ if $v_i = v'_i$ and $p_i$ otherwise, in which each voter $i$ is assigned a certain price $p_i$, whereas $B$ now limits the total costs we can invest for changing the votes. Regarding approval vote elections, they also considered Bribery' and \$Bribery' in which the price function depends on the number or, for the latter priced variant, the sum of the approval-specific prices of flipping approvals of the voter. Here, only the cost per voter is constrained by $B$, instead of the total sum. Thus, in the latter two definitions, we theoretically have the possibility to bribe every voter in the election simultaneously, which can be considered rather unnatural, while in the first definition we assume that the voters do not care how we change their votes as long as we pay the initial price, even if we only want to make small changes that may not be even very relevant to them. Faliszewski et al. [2009b] introduced a new variant called Microbribery for elections in which preferences consist of complete pairwise comparisons over the candidates but, unlike for the ordinal preference we introduced earlier, do not have to be transitive. In this variant, the agent pays a price of 1 for each comparison flip in a voter's preference, analogous to Bribery', but as in Bribery we bound the total costs. This more natural way of aggregating the costs, which is now standard in the literature, combines the advantages of the two previous definitions considered above: it is both sensitive to the strength of change in each preference and to the total costs. However, there are two major drawbacks with the definition of Microbribery: first, the preference model is rather unnatural and second, each flip of a pairwise comparison costs the same. These two points of criticism were addressed by Elkind et al. [2009] in the definition of Swap-Bribery, which is defined on ordinal preferences and also allows voters to set prices for swapping two adjacent candidates. In addition, they also defined Shift-Bribery in which only swaps moving the preferred candidate forwards, or in the destructive case the unpreferred candidate backwards, are allowed. Now, one reason for the success of these models compared to the general framework is the encoding of the price functions. In the general framework, the price function for each voter has to be encoded by $m!$ values for ordinal votes and $2^m$ values for approval votes, which makes an evaluation of the computational complexity of the problems on a polynomial scale pointless. However, this is not the case for the price functions in the previously mentioned models. Bribery, Bribery', and Microbribery do not require any parameters at all, while \$Bribery requires only one per voter, and \$Bribery' only $m$ per voter. Swap-Bribery requires $\mathcal{O}(m^2)$ per voter and Shift-Bribery only $\mathcal{O}(m)$.

Due to the massive number of results on bribery in the literature, we will limit ourselves here to the results that are most relevant to our later investigations in Chapter 3 and Chapter 5. Thus, we focus on the results regarding scoring rules and the two most prominent variants of bribery, namely Bribery and Swap-Bribery. Bribery is known to be efficiently decidable for plurality and veto (Faliszewski et al. [2006]), 2-approval and 2- and 3-veto (Lin [2012]) and NP-complete for $k$-approval with $k \geq 3$ and $k$-veto with $k \geq 4$ (Lin [2012]), Borda (Brelsford et al. [2008]), STV (Xia [2012]), Copeland (Faliszewski et al. [2009b]), and many others. In particular, it is known that Bribery is NP-complete for all pure scoring rules apart from the previously mentioned ones and $(2, 1, \dots, 1, 0)$ for which it is in P, whereby for the former probably no polynomial-time solution exists (Hemaspaandra and Schnoor [2016]). Similar results are available for Swap-Bribery, for which Elkind et al. [2009] have shown that there exists a polynomial-time algorithm for plurality and veto. For all other pure scoring rules, the problem is NP-complete by a simple reduction from Possible Winner which, according to Betzler and Dorn [2010] and Baumeister and Rothe [2012], is NP-complete for all pure scoring rules except plurality and veto. It should be mentioned that besides the worst-case results mentioned above, the problems were also studied in terms of their approximability, parameterized complexity, and complexity on restricted preference domains, e.g., by Brelsford et al. [2008], Dorn and Schlotter [2012], Bredereck et al. [2016], and Faliszewski et al. [2019, 2021], Knop et al. [2020], and Elkind et al. [2020], among others.

In the following, we will discuss further models of bribery in the literature. More recently, Knop et al. [2020] considered a generalization and combination of swap bribery with election control, where the agent can pay for swaps as well as, in the case of approval voting rules, for a shift in the approval threshold of a voter, i.e., the position in his/her ordinal preference at which he/she starts approving candidates, as well as for removing and adding voters as in the respective control problems. Moreover, they have determined the long unresolved parameterized complexity of swap bribery, in terms of the number of candidates using $n$-fold integer programs. Previously, Obraztsova and Elkind [2012] considered an extension of the manipulation problem, called optimal manipulation, in which the manipulator tries to achieve a manipulation that deviates as little as possible from his initial vote with respect to the unweighted swap, the footrule distance, or the maximum displacement distance. Thus, the problem corresponds to bribery restricted to a single bribable voter. The motivation is that even if the manipulator is willing to change his or her vote, he or she does not want to deviate too much from his or her initial vote due to convictions or fear of being discovered, especially if the votes are public. The distance restricted bribery model of Yang et al. [2016] combines the above-mentioned Bribery and Bribery' model of Faliszewski et al. [2006, 2009a] differently than the definition of Microbribery. Instead of looking at the total cost, the total number of bribable voters is still restricted and in addition there is a threshold for how much a bribed vote may deviate from the initial vote according to a certain distance. The motivation for introducing the vote-wise distance restriction is the same as for optimal manipulation, whereas here not only the voters want to hide their dishonesty, but also the agent may want to hide the whole bribery attempt. Similar to the work of Yang et al. [2016], Dey [2019] also studied Bribery' and \$Bribery with additional distance restrictions. Another interesting extension of \$Bribery was introduced by Dey et al. [2016], the so-called frugal bribery. Here, the very natural restriction is imposed that the agent only approaches voters who also prefer the distinguished candidate of the agent over the current winner and are thus presumably more willing to cooperate. Constructive and destructive bribery with respect to distances as price functions, in the fashion of Swap-Bribery, with the restriction on total costs, has been studied by us, which we discuss in much more detail in Chapter 3 and in Chapter 5.

As mentioned in the introduction to this section, the lines between malicious bribery and the generally considered acceptable, and in some cases even positive, promotion of one's own agenda are easily blurred. Hence, the underlying problem of bribery has also been examined in other contexts. Note that all the problems mentioned in the following can be formulated as bribery problems. For example, Christian et al. [2007] examined a variant of bribery in the sense of lobbying. Further, Elkind and Faliszewski [2010] studied shift bribery as a framework for campaigning and, recently, Zagoury et al. [2021] have studied negative campaigning in a slightly different model where the possible degradation operations are given as a set. However, even if these problems have a different motivation than bribery, they are still interference problems in which an agent tries to influence the outcome of an election.

On the other side, there are several concepts in the literature where bribery only serves as an underlying framework, but the goal is not interference but analysis. The best known is the margin-of-victory (MoV) problem, which asks how many votes must be replaced at least for the winner of the election to change. Getting back to our bribery models, this problem is equivalent to the destructive variant of Bribery. The main motivation here is to assess the strength of a candidate's victory and, related to that, the robustness of the election outcome. Thus, a high MoV speaks for such a clear election outcome that even the possibility of minor interferences in the conduct of the election does not cast doubt on the legitimacy of the result. This problem has been studied most notably by Cary [2011], Magrino et al. [2011], and Xia [2012]. Note that while many of the election problems are rather theoretical, the consideration of the MoV is often applied in practice and is either common practice or even incorporated in the law. For an overview regarding the legislation concerning U.S. elections see e.g., Tokaji [2004]. Following the same motivation, Shiryaev et al. [2013] investigated the destructive variant of swap bribery with unit costs, referred to as the robustness problem. Compared to the MoV, the robustness problem allows for a more fine-grained evaluation of the robustness of an election outcome by taking into account errors, manipulations, and uncertainties within votes. As a next step regarding the study of robustness, the complexity of the counting variants of the respective problems were examined by Boehmer et al. [2021a] and also by us. We will discuss those attempts in more detail in Chapter 5.

So far, we have focused mainly on bribery regarding single-winner elections, whereas bribery was also investigated with respect to multi-winner elections by Faliszewski et al. [2017] and with respect to iterative elections by Maushagen et al. [2018] and Zhou and Guo [2020]. Robustness was also studied regarding multi-winner elections by Bredereck et al. [2017, 2021].

Finally, it must be mentioned that bribery is not only studied in the context of elections but also in related areas. For example, Baumeister et al. [2011a] studied the problem of bribery regarding judgment aggregation, Rey and Rothe [2011] regarding path-disruption games in graphs, Mattei et al. [2015] regarding sports tournaments, Grandi and Turrini [2016] regarding rating systems, Erdélyi et al. [2017, 2020] regarding group identification, D'Angelo et al. [2021]

regarding vote delegation games in liquid democracy, and Boehmer et al. [2021b] regarding the stable marriage problem.

## 2.3 Sports

### 2.3.1 Competitions & Tournaments

Besides elections we also consider competitions and tournaments, especially in the context of sports tournaments. As before for collective decision-making processes, the developments of the last decades have made the computational of problems related to sports tournaments more and more relevant. Those developments include, for example, the widespread establishment of online gaming and e-sports, the ongoing monetization of sports e.g., through advertising and gambling, and the amount of freely available data from countless sporting events. The reason why we consider tournaments here together with elections are the strong similarities in the basic framework. As with elections, we are given a set of agents from which a winner has to be determined. Adapted to the scenario, however, we do not refer to it as the set of candidates here, but as the set of *teams* $T = \{t_1, \dots, t_n\}$. Unlike elections, the winners here are not to be determined based on voters' preferences but based on performance comparisons between the teams. We distinguish between two different types of comparisons: rankings and pair-wise comparisons.

#### Ranking-Based Tournaments

*Rankings* are defined analogously to ordinal preferences in elections, that is all teams are ranked from position one to position $n$. Rankings can have different origins, the most obvious being race results as commonly used in motorsports or skiing, where participants are ranked according to their finishing time. However, rankings can also come from a jury, as in the Eurovision Song Contest, where each juror ranks the participants according to their perceived performance. While this approach is again very similar to voting, the rationale here is different. While in an election voters choose candidates according to their preferences, jurors are supposed to rank participants objectively according to their performance.

Competitions and tournaments regarding rankings are often evaluated using *scoring systems*, which are defined analogously to the scoring rules used in elections. Examples include skiing, which currently uses $\vec{\alpha} = (100, 80, 60, 50, 45, 40, 36, 32, 29, 26, 24, 22, 20, 18, 16, 15, 14, 13, 12, 11, 10, 9, 8, 7, 6, 5, 4, 3, 2, 1, 0, \dots, 0)$ in the major races such as the world cup, and the Formula 1, which has used many different scoring systems over the years, such as $\vec{\alpha} = (9, 6, 4, 3, 2, 1, 0, \dots, 0)$ in 1961–1990, $(10, 6, 4, 3, 2, 1, 0, \dots, 0)$ in 1991–2002, $(10, 8, 6, 5, 4, 3, 2, 1, 0, \dots, 0)$ in 2003–2009, and $(25, 18, 15, 12, 19, 8, 6, 4, 2, 1, 0, \dots, 0)$ in 2010–2018. However, scoring systems are also used in non-sports competitions such as the Eurovision Song Contest with $\vec{\alpha} = (12, 10, 8, 7, 6, 5, 4, 3, 2, 1, 0, \dots, 0)$. In Chapter 4 we show that the chosen scoring system can have a massive impact on the outcome, not only theoretically but also in practice.

In practice, there are also other approaches for evaluating ranking-based competitions. One example is the *Inverse-Borda-Nash* rule used in the Olympic climbing competition, which was named and studied by Kruger and Schneckenburger [2019].

#### Pair-Wise Tournaments

*Pair-wise comparisons*, on the other hand, represent the outcomes of direct comparisons between two teams, which we refer to as matches. The actual nature of these outcomes will be discussed in in the following, depending on the chosen tournament format.

We start by considering score-based pair-wise tournaments, following the very general definitions by Kern and Paulusma [2004]. In a *score-based pair-wise tournament*, the tournament $\mathcal{T} = (T, M)$ consists of the set of teams $T = \{t_1, \ldots, t_n\}$ with $n \geq 2$ and a set of pair-wise matches $M = \{m_1, \ldots, m_g\}$ with $g \geq 1$. As described above, each match $m_i$ in $M$ is associated with a tuple $(t_x, t_y)$ of competing teams $t_x, t_y \in T$ with $t_x \neq t_y$. For the purpose of visualizing a tournament $\mathcal{T}$, we use the so-called *match graph* of $\mathcal{T}$, an undirected multi-graph consisting of the teams as nodes and an edge for each match connecting the participating teams. Over the course of a tournament, the matches result in specific outcomes which are associated with certain amounts of points that the teams receive for their performances. For example, in football (soccer) the winning team receives 3 points, while the other team receives 0 points. In the case of a draw, both teams receive 1 point. In basketball it is the same, except that the winning team gets 2 points instead of 3. In baseball there are no ties and either one or the other team receives a point. In volleyball there are several different combinations of points based on the strength of the win. Formally, we specify these rules by the set of possible *outcomes* $O = \{(\alpha_1, \beta_1), \ldots, (\alpha_\ell, \beta_\ell)\}$ with $\ell \geq 1$ and $\alpha_s, \beta_s \in \mathbb{N}_0$ for $1 \leq s \leq \ell$. In practice, some sports use rational amounts of points, such as in chess with $1/2$ points for a draw. However, those set of outcomes can be scaled to equivalent ones over $\mathbb{N}_0$. If a match $m : (t_x, t_y)$ results in an outcome $(\alpha_s, \beta_s)$, it means that team $t_x$ gets $\alpha_s$ points and team $t_y$ gets $\beta_s$ points. At the end of the tournament, the teams that have the maximum number of points are the winners. By default, we assume that there is at least one outcome $(\alpha_s, \beta_s) \in O$ with $\alpha_s \neq \beta_s$. In many cases, we will also assume that, as in most real-world tournaments, the set of outcomes is *symmetric*, that is $(\beta_s, \alpha_s) \in O$ for all $(\alpha_s, \beta_s) \in O$.

Note that for symmetric sets of outcomes the actual order of the teams in the match tuples is negligible from a conceptual point of view, as long as it is clear which team receives which amount of points. In many real-world tournaments, the first team of a match tuple is usually the so-called home team, i.e., the team in whose sports facility the match takes place. Accordingly, in this case, the second team is the away team that travels for the match. In general, the above definition of the set of outcomes allows, for example, the away team to receive more points than the home team in the case of a draw or a win, in order to counteract the home advantage, which has been statistically proven in many tournaments. However, the home advantage is overcome in most tournaments by the fact that each pair of teams always plays an even number of matches against each other, and thus each team can be the home team the same number of times. Furthermore, it should be noted that the above definition of a tournament also allows for extremely unrealistic and unfair tournaments, e.g., tournaments in which the teams play different numbers of matches. This leads us to the most common type of score-based pair-wise tournaments, but also of tournaments in general, used in practice.

A *round-robin tournament* is a score-based pair-wise tournament in which each teams plays exactly once against each other team. A round-robin tournament with $k$ rounds is a concatenation of $k$ single-round round-robin tournaments as defined above. We illustrate these definitions with an example.

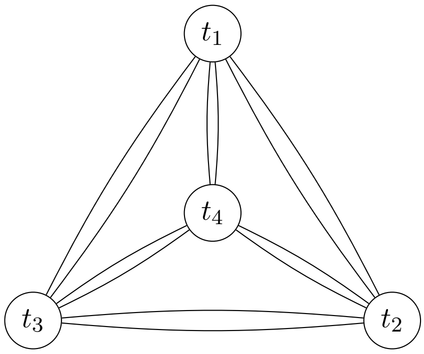


Figure 2.7: Match graph for the round-robin tournament $\mathcal{T}$ considered in Example 2.7.

**Example 2.7.** Consider the following two-round round-robin tournament $\mathcal{T} = (T, M)$ over the set of teams $T = \{t_1, t_2, t_3, t_4\}$ and the set of matches $M = \{m_1, \ldots, m_{12}\}$ given as follows.

$$\begin{array}{llll} m_1 : (t_1, t_2) & m_2 : (t_2, t_1) & m_3 : (t_1, t_3) & m_4 : (t_3, t_1) \\ m_5 : (t_1, t_4) & m_6 : (t_4, t_1) & m_7 : (t_2, t_3) & m_8 : (t_3, t_2) \\ m_9 : (t_2, t_4) & m_{10} : (t_4, t_2) & m_{11} : (t_3, t_4) & m_{12} : (t_4, t_3) \end{array}$$

The corresponding match graph is illustrated in Figure 2.7. We now assume that the tournament was conducted using the set of outcomes $O = \{(2, 0), (1, 1), (0, 2)\}$ and the matches resulted in the following outcomes.

$$\begin{array}{llll} m_1 \to (2, 0) & m_2 \to (1, 1) & m_3 \to (1, 1) & m_4 \to (2, 0) \\ m_5 \to (0, 2) & m_6 \to (1, 1) & m_7 \to (0, 2) & m_8 \to (1, 1) \\ m_9 \to (1, 1) & m_{10} \to (2, 0) & m_{11} \to (2, 0) & m_{12} \to (1, 1) \end{array}$$

To determine the winner(s), the scores of the teams are calculated by adding up the points over the individual matches. Thus, $t_3$ wins the tournament with 9 points, ahead of $t_4$ with 7 points, $t_1$ with 5 points, and $t_2$ with 3 points.

There are numerous real-world examples for sports using round-robin tournaments. We have summarized some of them together with the currently most commonly used set of outcomes in Table 2.1. Note that in this work we focus on stand-alone round-robin tournaments where the winning teams are the teams with the maximum number of points, possibly followed by a tie-breaking. On the other hand, in many sports it is common that after the round-robin tournament (or after multiple separated round-robin tournaments), often referred to as the *regular season*, a certain number of top-ranked teams participate in a tournament for the championship, often referred to as *post-season*.

In addition, round-robin tournaments are often used as intermediate or qualification phases in larger tournaments. Examples are the so-called group stages of the FIFA World Cup or the UEFA Champions League, in which a certain number of top-ranked teams from each group then compete against each other in an elimination tournament. In addition, there are many variations of round-robin tournaments in practice. For example, in the major North American leagues, namely the National Football League, the National Basketball Association, the National Hockey League, and the Major League Baseball National League and American League, teams are divided into several divisions and smaller regional groups. The teams

| *Sport* | *Set of Outcomes $O$* |
|---:|---|
| Baseball | $\{(1,0),(0,1)\}$ |
| Chess | $\{(1,0),(1/2,1/2),(0,1)\}$ |
| Football (Soccer) | $\{(3,0),(1,1),(0,3)\}$ |
| Ice Hockey | $\{(3,0),(2,1),(1,2),(0,3)\}$ or $\{(2,0),(2,1),(1,2),(0,2)\}$ |
| Volleyball | $\{(3,0),(2,1),(1,2),(0,3)\}$ |

Table 2.1: Examples of commonly used sets of outcomes in sports.

usually play a multi-round round-robin tournament in the regional group and in addition play several games against teams from other groups and other divisions.

One reason why round-robin tournaments are so popular is the relative fairness and, related to that, the perceived closeness of the final standing to the actual performance level. The main reasons for this are that there is comparatively low dependence of the outcome on the often randomly determined circumstances, that each team actually competes directly or even several times against each competitor, and that the high number of matches compensates for fluctuations in performance over the season. However, this comes at the cost of a very high number of matches, possibly including even irrelevant matches at the end.

In a *single-elimination tournament* (also known as cup tournament), teams compete in rounds, with each team that has not yet been eliminated competing in one match per round. The losing teams in each round are eliminated from the tournament. We illustrate a single-elimination tournament in Figure 2.8. Well-known examples for this type of tournaments are the final stage of the FIFA World Cup or the UEFA Champions League, where in the latter the teams actually compete against each other twice, except for the final, and the goals are summed up in order to overcome a possible home advantage. Especially compared to the round-robin tournaments, two main advantages are that we only need a relatively small number of matches and that every remaining match is relevant and can have a direct impact on the outcome. However, single-elimination tournaments have one major drawback: the difficulty of a team to win the tournament can be highly dependent on the initial drawing of the matches, the so-called seeding. This can massively affect perceived fairness and sometimes leads to the conclusion that the winner was not necessarily the best team, but was lucky, either in the seeding or in the matches.

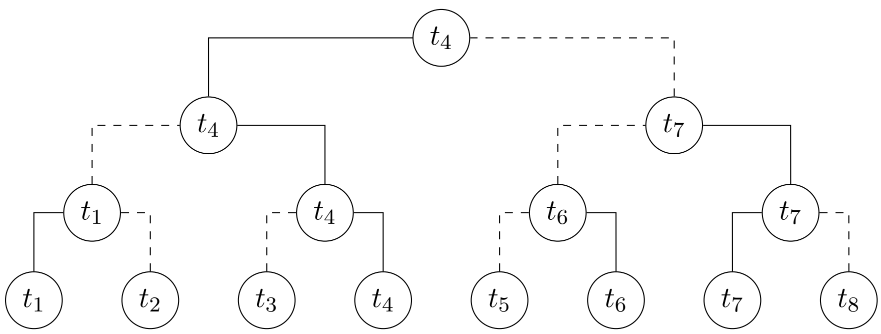


Figure 2.8: Example of a single-elimination tournament over eight teams with winner $t_4$.

There are also other tournament formats in practice, some of which are extensions and combinations of the above. For example, there are also *double-elimination tournaments*, in which a team is only eliminated if it has lost twice and is moved to a second branch of the tournament after the first defeat. The *Swiss system*, often used in chess and becoming more and more popular, is a combination of a score-based pair-wise tournament and a round-based elimination tournament, in which the teams earn points according to a set of outcomes and compete in each round against a team with a similar score.

### 2.3.2 Computational Problems

As for elections, we will now present the computational problems for tournaments that are relevant to our research here. In particular, we will focus on the problems regarding round-robin tournaments, which we will later study in Chapter 6.

#### Scheduling

A key aspect of organizing a tournament is scheduling the matches. Before it comes to choosing an exact date and time, there is usually the question of which matches will take place in parallel or in the same time frame, the so-called *matchday*. It is usually assumed by default that each team participates in at most one match per matchday. Now, if we look at a score-based pair-wise tournament $\mathcal{T} = (T, M)$, we can see that such a schedule $S : M \to D$ with $S(m) \neq S(m')$ for two adjacent matches $m, m' \in M$ with $D = \{1, \ldots, d\}$ denoting the set of available matchdays is actually equivalent to an edge coloring of the match graph.

**Example 2.8.** Consider the two-round round-robin tournament $\mathcal{T}$ given in Example 2.7. One possible schedule $S$ with the minimum number of matchdays required $d = 6$ is the following.

| Matchday 1 ($S(m) = 1$) | Matchday 2 ($S(m) = 2$) | Matchday 3 ($S(m) = 3$) |
|---|---|---|
| $m_5 : (t_1, t_4)$ | $m_3 : (t_1, t_3)$ | $m_1 : (t_1, t_2)$ |
| $m_7 : (t_2, t_3)$ | $m_{10} : (t_4, t_2)$ | $m_{11} : (t_3, t_4)$ |

| Matchday 4 ($S(m) = 4$) | Matchday 5 ($S(m) = 5$) | Matchday 6 ($S(m) = 6$) |
|---|---|---|
| $m_6 : (t_4, t_1)$ | $m_4 : (t_3, t_1)$ | $m_2 : (t_2, t_1)$ |
| $m_8 : (t_3, t_2)$ | $m_9 : (t_2, t_4)$ | $m_{12} : (t_4, t_3)$ |

Figure 2.9 shows an edge coloring of the match graph of $\mathcal{T}$ corresponding to $S$.

Due to this equivalence, we already know from the results of Holyer [1981] regarding the edge coloring of cubic (3-regular) graphs that checking whether a score-based pair-wise tournament can be scheduled with a given number of available matchdays is NP-complete, even if there are only three matches per team and the number of available matchdays is three. On the positive side, we also know, through the theorem of Vizing [1964], that if $\Delta$ denotes the maximum number of matches of a team in the tournament, the tournament can be scheduled using $\Delta + 1$ matchdays.

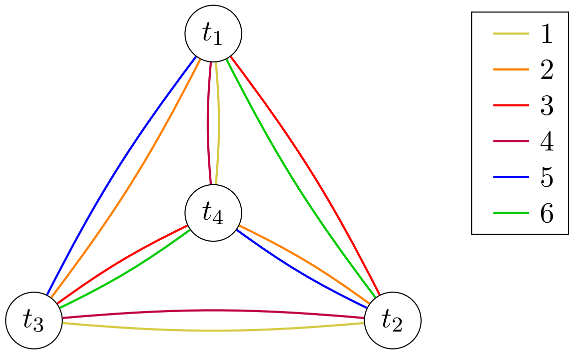


Figure 2.9: Edge coloring of the match graph of tournament $\mathcal{T}$ corresponding to the schedule $S$ given in Example 2.8.

For the special case of single-round round-robin tournaments, on the other hand, the problem can be solved efficiently as the match graph is a complete graph which can be efficiently colored using the minimum number of colors required, namely $\Delta$ for even $n$ and $\Delta+1$ for odd $n$ with $\Delta = n-1$. For multi-round round-robin tournaments, schedules for the individual rounds are usually concatenated. There are numerous methods that can be used to find such a schedule with the minimum number of matchdays required. Since most methods are designed for an even number of teams, in the case of an odd number of teams a placeholder team is usually added, whose matches simply correspond to a break for the respective team. The standard method is the so-called circle method, which is also known as the canonical 1-factorization in the context of graphs. However, in practice there are a number of other methods that take into account other aspects, such as teams alternating between home and away matches as consistently as possible, a team not having to play several matches in a row against the top teams, and that the schedule is randomized to some extent for fairness reasons. For a comprehensive overview on the various aspects of round-robin tournament scheduling, we refer to the survey by Rasmussen and Trick [2008].

An interesting question related to scheduling round-robin tournaments is whether one can simply schedule the matchdays one after the other as desired and achieve a valid schedule with the minimum number of matchdays required. Rosa and Wallis [1982] have shown that this is not the case and have studied the property of partial schedules over only a subset of matchdays of not being extendable to complete schedules with only the minimum number of matchdays required. They referred to this property of a partial schedule, or equivalently of a partial edge coloring, as *premature*. An example for such a premature schedule is given in Figure 2.10. Furthermore, Colbourn [1983] showed that checking whether a given partial schedule is non-premature is NP-complete. The results of Rosa and Wallis [1982] and Colbourn [1983] show that scheduling round-robin tournaments is actually not trivial and especially not if we would like the schedule to fulfill certain requirements. This means in particular that one cannot simply schedule certain match days as desired, for example to increase television revenues by having as few top matches as possible in parallel. Moreover, these observations are of immense theoretical importance, especially for us, since they show that the requirement of a schedule, as they exist for real-world tournaments, is a non-trivial restriction with respect to complexity results. This is especially relevant since most previous results did not involve schedules and the instances constructed in the reductions could not occur in practice.

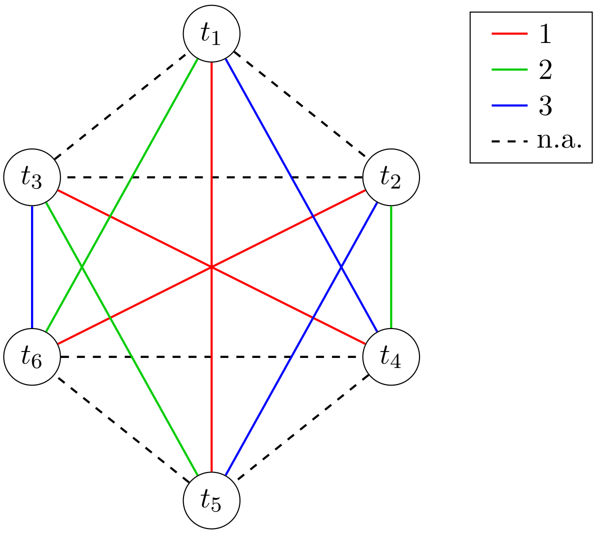


Figure 2.10: Premature schedule for a single-round round-robin tournament $\mathcal{T} = (T, M)$ over the teams $T = \{t_1, \ldots, t_6\}$.

### Elimination

In the course of a tournament, the question of who will or can be the winner inevitably arises at some point. For single-elimination tournaments, this question is trivial: any team that has not yet been eliminated can still become the champion. For round-robin tournaments and score-based pair-wise tournaments in general, this question is not so easy to answer. We illustrate the problem in the following example.

**Example 2.9.** Suppose we consider a partially played single-round round-robin tournament $\mathcal{T} = (T, M)$ over $T = \{t_1, t_2, t_3, t_4, t_5, t_6\}$ with $O = \{(3, 0), (1, 1), (0, 3)\}$ and the following matches, schedule, and results.

| Matchday 1 | Matchday 2 | Matchday 3 |
|---|---|---|
| $m_1 : (t_1, t_6) \to (1, 1)$ | $m_4 : (t_1, t_5) \to (3, 0)$ | $m_7 : (t_1, t_4) \to (3, 0)$ |
| $m_2 : (t_2, t_5) \to (3, 0)$ | $m_5 : (t_2, t_3) \to (3, 0)$ | $m_8 : (t_2, t_6) \to (3, 0)$ |
| $m_3 : (t_3, t_4) \to (1, 1)$ | $m_6 : (t_4, t_6) \to (0, 3)$ | $m_9 : (t_3, t_5) \to (0, 3)$ |

| Matchday 4 | Matchday 5 |
|---|---|
| $m_{10} : (t_1, t_3)$ | $m_{13} : (t_1, t_2)$ |
| $m_{11} : (t_2, t_4)$ | $m_{14} : (t_3, t_6)$ |
| $m_{12} : (t_5, t_6)$ | $m_{15} : (t_4, t_5)$ |

The scores after the first three matchdays are as follows: $t_2$ leads with 9 points, followed by $t_1$ with 7 points, $t_6$ with 4 points, $t_5$ with 3 points, $t_3$ with 1 point, and $t_4$ with 1 point.

We now want to check which teams can still become the unique winner of the tournament. The teams $t_3$, $t_4$, and $t_5$ are already eliminated in this respect, as neither of them can surpass team $t_2$ with at most 6 remaining points. Clearly, the current leading team $t_2$ can become the unique winner by winning its remaining two matches, regardless of the outcomes of the other matches. The same is true for team $t_1$, which will definitely be the unique winner if it wins its two remaining matches, since one of them is against $t_2$. The third team $t_6$, however, has no chance to become the unique winner, although it can still surpass team $t_2$ with 6 additional points. The key problem for $t_6$ is that $t_1$ and $t_2$ are still facing each other and $t_2$ has to lose this match so that $t_6$ can surpass it. However, in that case $t_1$ has at least 10 points and $t_6$ cannot become the unique winner anymore.

While in this small example the reasoning why a certain team can no longer become unique winner, although it can still catch up with the leading team, was quite straightforward, the reasoning for larger examples can become very complex. This leads us to the formal definition of the corresponding computational problem. Formally the problem of checking whether a certain team can still become a unique winner of a partially played tournament is referred to as the *sports elimination problem*. Given a fixed set of possible outcomes $O$, the decision problem is given as follows.

| $O$-ELIMINATION | |
|---|---|
| **Given:** | A score-based pair-wise tournament $\mathcal{T} = (T, M)$, a partial assignment of outcomes in $O$ to $M$, and a distinguished team $p \in T$. |
| **Question:** | Is there a completion of the partially played tournament $\mathcal{T}$, such that $p$ ends up as the unique winner? |

In fact, the sports elimination problem is the counterpart to the possible winner problem for elections that we discussed in Section 2.2.

Schwartz [1966] has shown that the elimination problem for baseball with $O = \{(1,0),(0,1)\}$ can be solved efficiently, by reformulating it as a network flow problem with $\mathcal{O}(n^2)$ nodes and applying the algorithm by Ford and Fulkerson [1956], whereas $n$ denotes the number of teams. Gusfield and Martel [1992] later improved this result by finding a formulation using $\mathcal{O}(n)$ nodes. Subsequently, Wayne [2001] showed that there is a uniform score value for all teams that can be calculated in polynomial time and for which a team is eliminated if and only if it can no longer reach it. Bernholt et al. [1999], on the other hand, showed that the elimination problem is actually NP-complete for $O = \{(3,0),(1,1),(0,3)\}$, even in the case where each team has at most three remaining matches and the tournament is an ordinary, partially played round-robin tournament. They have also shown that the problem can be solved efficiently in the case that each team has at most two remaining matches. However, they did not pay attention to the existence of a suitable schedule that would allow the constructed situations to occur in a real-world tournament. We will discuss this issue later in Chapter 6.

Kern and Paulusma [2004] studied a more general variant of the elimination problem for score-based pair-wise tournaments in which the current scores are given as a vector and not by a partially played tournament. They presented a scheme to normalize set of outcomes to show the following dichotomy result. For all sets of outcomes $O = \{(j, k-j) \mid 0 \leq j \leq k\}$ for $k \in \mathbb{N}$ and equivalent ones, they showed that the problem is solvable in polynomial time by extending the approach by Schwartz [1966]. For all other sets of outcomes, the problem is NP-complete, even if each team has at most three matches remaining. However, this result should be taken with a grain of salt: the tournaments constructed in the reduction are for the most part very unrealistic and the initial score vector can be set arbitrarily, even if the score vector cannot be generated by a partially played pair-wise tournament[3]. The parameterized complexity of this variant of the elimination problem was also later studied in much more detail by Cechlárová et al. [2016], with respect to various parameters such as the number of remaining matches per team and graph parameters, such as the tree width and the minimum feedback arc set size, of the match graph regarding the remaining matches. While efficiency results and parameterized efficiency results transfer from this more general variant to the normal variant with a partially played tournament, the same applies in the other direction for hardness results.

The elimination problem has also been studied for other types of tournaments. As mentioned at the beginning, for elimination tournaments, a team can still become champion if it has not yet been eliminated. Neumann and Wiese [2016] studied the complexity of the elimination problem with respect to the system used in debating leagues, where four teams compete against each other, and with respect to the Swiss system. In addition, many variants and extensions of the elimination problem have been studied. Hoffman and Rivlin [1967] studied an extension of the elimination problem for Baseball where the goal is to check whether a team can still reach at least a certain position. The motivation here is for example to check if a team can still qualify for the post-season, a promotion, or for international tournaments. McCormick [1999] showed that this problem is NP-complete for Baseball with $O = \{(1, 0), (0, 1)\}$, for which the regular elimination problem can be solved efficiently. Mattei et al. [2015] studied the probabilistic variant of the elimination problem, where one not only wants to know whether a team can become champion, but also with what probability it will become champion. They studied this problem and various interference problems such as manipulation and bribery for different tournament formats, including round-robin tournaments and single-elimination tournaments. We will discuss this variant in much more detail in Chapter 6.

Another very interesting problem in connection with tournaments, more specifically single-elimination tournaments, is the problem of agenda control, where the goal is to construct a single-elimination tournament in such a way that, under certain assumptions, the victory of a team is guaranteed or the chance of victory is maximized. A special case of this problem is the tournament fixing problem, where the structure of the tournament tree is given, and an advantageous initial seeding has to be found. We refer to the survey by Williams [2016] for a comprehensive overview.

[3] In fact, Pálvölgyi [2009] showed that the problem of checking whether a given score-based pair-wise tournament can generate a certain score vector is itself already NP-complete for many sets of outcomes, as already suspected by Kern and Paulusma [2004].

# Chapter 3

## Distance Bribery in Elections

This chapter deals with the classical computational decision complexity of constructive and destructive bribery with respect to distance-based price functions in scoring rule elections.

### 3.1 Summary

Inspired by the swap bribery problem by Elkind et al. [2009] and the robustness problem by Shiryaev et al. [2013], i.e., destructive unit cost swap bribery, we study the problem of constructive and destructive bribery in scoring rule elections on ordinal preferences in which the price functions are given by distances, namely the weighted swap distance and the weighted footrule distance as introduced by Kumar and Vassilvitskii [2010][1]. For a set of candidates $C$ the *weighted swap distance* and the *weighted footrule distance* between two ordinal preferences $v$ and $v'$ over $C$ are defined as follows:

$$\mathrm{swap}^w_\pi(v,v') = \sum_{\substack{x >_v y \text{ and } y >_{v'} x,\\ x,y\in C}} \pi(x,y)$$

$$\mathrm{fr}^w_\pi(v,v') = \sum_{y\in C}\left|\sum_{x>_v y,\ x\in C}\pi(x,y) - \sum_{x>_{v'} y,\ x\in C}\pi(x,y)\right|$$

with $\pi : C \times C \to \mathbb{N}_0$ with $\pi(x,y) = \pi(y,x)$ for $x,y \in C$ with $x \neq y$ and $\pi(x,x) = 0$ for $x \in C$ denoting the *weight function* of a given voter. Note that distances, by satisfying the (pseudo)metric properties, appear to be a rather natural framework for price functions of rational voters, both for the study of bribery and robustness. In particular, the original bribery problems presented in Section 2.2.3 use vote-wise distances as price functions. Namely, Bribery uses the discrete distance and \$Bribery uses a weighted variant of the discrete distance, Bribery' and \$Bribery' use the regular and a weighted variant of the Hamming distance, respectively, Microbribery uses the Hamming distance on the pairwise comparisons, and Swap-Bribery and Shift-Bribery use the weighted swap distance. Especially, also the robustness motivated problems like the one of Shiryaev et al. [2013] and the margin-of-victory problem use the swap distance and the discrete distance, respectively.

In particular, we are interested in understanding how the degree of expressiveness of the price functions affects the complexity of the problem. Thus, we scale the expressiveness of the price functions by controlling the adjustability of the pairwise weights $\pi(x,y)$ for $x,y \in C$ with $x \neq y$ that parameterize the distances, following Kumar and Vassilvitskii [2010]. We distinguish between the *unweighted* variants with $\pi(x,y) = 1$, the *element-weighted* variants $\pi(x,y) = \varphi(x)\cdot\varphi(y)$ with candidate weights $\varphi : C \to \mathbb{N}_0$, and the *fully weighted* variants

[1] Kumar and Vassilvitskii [2010] actually used the terms *generalized* swap/footrule distance instead of *weighted* swap/footrule distance, hence the name of theirs and our publication.

where $\pi(x, y) = \pi(y, x)$ can be set arbitrarily. In fact, the expressiveness increases in the order in which the variants are given, in the sense that each variant generalizes the previous ones. However, the required amount of information also increases, noticeable in the number of parameters, namely $\mathcal{O}(1)$, $\mathcal{O}(m)$, and $\mathcal{O}(m^2)$. All three variants are easy to motivate. In a setting like the one of Shiryaev et al. [2013], where we only know the preferences of the voters, the most intuitive way without making any further, possibly too speculative, assumptions is to weight all pairs of candidates equally. This is particularly appropriate if we want to measure the robustness and the uncertainties arise from technical errors in the collection or transmission of the election data and are not dependent on the candidates. While this model is appealing for its relatively low information demand and general complexity, in many cases it is not realistic enough to represent voters' natural sense of the strength of changes and deviations in their preferences. For example, a voter is likely to have a relatively strong opinion regarding the placement of certain candidates, such as his or her favorite and least favorite candidates, while the placement of other candidates are relatively unimportant to him or her as long as they do not interfere with the previously mentioned ones. This can be reflected in the element-weighted variant, where each voter can weight the candidates, e.g., according to their personal perception of importance. While this model is already relatively expressive, there are also natural settings that cannot be represented here. For example, it is possible that a voter cares very much that a certain group of unpopular candidates is and remains on the last positions, but a change in the intern positions of this group makes no real difference to him or her. This is where the fully weighted variant would come into play, which requires a lot of information, but is also highly adjustable.

Our results, together with the previous results, cover almost the entire spectrum for the different degrees of expressiveness, for the constructive and destructive case, and for the most prominent scoring rules. We were able to show that the transition in complexity, from P to NP-completeness, depending on the voting rule, distance, and constructive or destructive case, can happen at any point of transition to the next more expressive weighting: For plurality, for example, all cases are in P, while for $(2, 1, \ldots, 1, 0)$ in the destructive case the footrule is still in P for the element-weighted variant, but NP-complete for the fully weighted one. For $k$-approval with $k \geq 2$, the problem is still in P for the constructive case and the unweighted swap distance but is already NP-complete for the element-weighted variant. Finally, for Borda in the constructive case, the problem is already NP-complete for the unweighted variants of the two distances. Thus, we have found a rich and diverse spectrum of complexity results depending on the expressiveness of the weighting and the price function.

We were also able to develop much more general results. For example, our efficiency result for the destructive case and the fully weighted variant for both distances covers not only the $k$-approval voting rules with fixed $k$ including plurality and veto, but pure scoring rules in general whose scoring vectors behave rather statically. On the other hand, our hardness result covers many scoring rules whose score vectors grow more dynamically, such as Borda, $\lfloor m/2 \rfloor$-approval, and many others. Furthermore, for the constructive case for the element-weighted variants, we could develop a complete dichotomy result covering all pure scoring rules. In addition, our hardness results cover many additional restrictions. For example, our dichotomy result also applies to uniform candidate weights for all voters with only a small set of total weights necessary. The results for the destructive case, which also cover the constructive case, hold even if only one particular voter is bribable at all, which corresponds to the optimal manipulation problem of Obraztsova and Elkind [2012].

Recently, Chakraborty and Kolaitis [2021] studied the possible winner problem with preferences restricted to partial chains, i.e., partial orders consisting of total orders on a non-empty subset of candidates, which, unlike the normal possible winner problem, can be easily reduced to the element-weighted swap distance bribery by simply setting the weights for the candidates in the subset to 1, for all others to 0, and setting the distance limit to 0 as well. Their dichotomy result with respect to pure scoring rules thus also confirms the dichotomy result here, even if restricted to candidate weights 0 and 1. However, the uniform weight functions for all voters are not covered, since the results for the case with a uniform set of candidates, which is equivalent to the possible winner problem with new candidates introduced by Chevaleyre et al. [2010], are not complete regarding pure scoring rules.

We answer a number of open questions in the literature, e.g., one by Shiryaev et al. [2013] concerning the complexity of destructive swap bribery on (pure) scoring rules and the complexity of robustness with respect to the footrule distance, and of Obraztsova and Elkind [2012] concerning the optimal manipulation problem for Borda and the weighted swap distance. Further, our results were later used for investigations on bribery and robustness in counting and probabilistic settings, e.g., by Boehmer et al. [2021a] and also by us, which we will discuss later in Chapter 5.

## 3.2 Publications

This work was published as:

> D. Baumeister, T. Hogrebe, and L. Rey. Generalized Distance Bribery. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, pages 1764–1771. AAAI Press, 2019.

A preliminary version of this publication appeared in the proceedings of the *7th International Workshop on Computational Social Choice (COMSOC 2018)*:

> D. Baumeister, T. Hogrebe, and L. Rey. Generalized Distance Bribery. In *Proceedings of the 7th Workshop on Computational Social Choice*, 2018.

## Personal Contribution

The writing and development of the model was done jointly with Dorothea Baumeister and Lisa Rey. The initial technical results were contributed by me. Parts of the results also appear in my master's thesis, which refers to the preliminary version (Baumeister et al. [2018]). Specifically, preliminary versions of Theorem 4, Theorem 10, Theorem 15, and Theorem 16 in Baumeister et al. [2019] first appeared in my master's thesis.

# Chapter 4

## Manipulative Design of Scoring Systems

This chapter deals with the classical computational decision complexity, the parameterized complexity, and the complexity of approximation in the context of designing and optimizing a scoring system that guarantees a desired outcome.

## 4.1 Summary

In the original possible winner problem as introduced by Konczak and Lang [2005], we are given partial preferences and want to check whether a particular candidate is a winner in one of the possible completions of the preferences with respect to a given voting rule. In the variant of the problem considered here, there is the twist that we have complete preferences, but our voting rule, here in particular our scoring system (scoring rule), is not completely specified. This problem has so far only been studied by Baumeister et al. [2011b] with respect to Copeland$^{\alpha}$ with adjustable tie factor $\alpha \in \mathbb{Q} \cap [0, 1]$ and scoring systems in a very restricted special case.

For the scoring systems considered here, the problem is formally given as follows.[1]

| Scoring System Existence | |
|---|---|
| **Given:** | A set of candidates $C$ with $\|C\| = m$, a list of profiles $V_1, V_2, \ldots, V_N$ over $C$, and a distinguished candidate $p \in C$. |
| **Question:** | Is there a scoring vector $\vec{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_m) \in \mathbb{Q}^m_{\geq 0}$ with $\alpha_m = 0$, such that $p$ is the unique winner of election $E_j = (C, V_j)$ with respect to $\vec{\alpha}$ for each $1 \leq j \leq N$? |

Note that compared to the version of Baumeister et al. [2011b], we consider a list of multiple profiles instead of a single profile, which we will explain later. However, all of our results also apply to the special case of a single profile. Note that while in the problem definition a single candidate $p$ is supposed to be the winner with respect to all profiles, by renaming the candidates in the profiles one can easily extend the problem to different candidates. In addition, we have introduced and studied a variant in which the possible solution $\vec{\alpha}' \in \mathbb{Q}^m_{\geq 0}$ may deviate only by at most a given value $K \in \mathbb{Q}_{\geq 0}$ from a given vector $\vec{\alpha} \in \mathbb{Q}^m_{\geq 0}$ with respect to a distance $\mathcal{D}$. We denote this variant by $\mathcal{D}$-Close Scoring System.

[1] In our original publications (see Baumeister and Hogrebe [2019b,a]), we specified that $\vec{\alpha} \in \mathbb{R}^m_{\geq 0}$ and argued that if a solution $\vec{\alpha} \in \mathbb{R}^m_{\geq 0}$ exists, then also a rational solution $\vec{\alpha} \in \mathbb{Q}^m_{\geq 0}$ exists, whereby we can focus on $\vec{\alpha} \in \mathbb{Q}^m_{\geq 0}$ without loss of generality. While this is true and the following issue does not affect the given results, one must be careful here to define the set of solutions over real-valued vectors. The reason is that verification problems with arbitrary vectors may become tricky, due to the implicit problem of comparing real numbers, of which the decidability and complexity strongly depends on the encoding.

We have studied Scoring System Existence based on several possible applications. In the context of elections, we can model the situation where an agent has predicted a collection of most likely profiles and now wants to propose a scoring rule that guarantees the victory of a particular candidate. Even though we use the term 'manipulative' in the title, the agent may also try to find a system that seems particularly fair or that satisfies certain properties. Another application closely related to elections are surveys and studies where participants are asked to rank a list of objects, candidates, etc. and the agent subsequently tries to find a system that supports his or her hypothesis or desired outcome. If a system already is in place, it is reasonable to aim at modifying the system as little as possible while trying to guarantee a certain outcome, which is the motivation for including the distance restriction in $\mathcal{D}$-Close Scoring System. This is especially relevant for us if we consider the application in sports and competitions, such as racing, skiing, the Eurovision Song Contest, and many others, where the participants receive points regarding their position in the competition. In addition to the motivation of ensuring an outcome with as little change as possible, the problem is particularly useful for evaluating the robustness of the outcome of an election or competition with respect to the chosen scoring system. Note that while the problems appear to be somewhat speculative, every voting rule and every system of deciding winners for competitions has been determined at some point and usually by a small group of individuals, and it would be particularly naive to assume that those same individuals decided free of any underlying motives. We illustrate the two problems in the following example.

**Example 4.1.** Consider the election $E = (C, V)$ with $C = \{a, b, c, d\}$ and $V = (v_1, v_2, v_3, v_4, v_5)$ with

$$\begin{array}{ll} v_1: & b > a > d > c \\ v_2: & c > a > b > d \\ v_3: & c > a > b > d \end{array} \qquad \begin{array}{ll} v_4: & d > c > b > a \\ v_5: & a > d > b > c \end{array}$$

from Example 2.5 and scoring system $\vec{\alpha} = (\alpha_1, \alpha_2, 1, 0)$. The scores of the candidates are given by $score(a) = \alpha_1 + 3 \cdot \alpha_2$, $score(b) = \alpha_1 + 4$, $score(c) = 2 \cdot \alpha_1 + \alpha_2$, and $score(d) = \alpha_1 + \alpha_2 + 1$. In Figure 4.1 we illustrate the space of scoring vectors with respect to $\alpha_1$ and $\alpha_2$. First, using Scoring System Existence, for each of the three candidates $a$, $b$, and $c$, we would find at least one integer vector in which the respective candidate is the unique winner, e.g. $(3, 2, 1, 0)$ for $a$, $(2, 1, 1, 0)$ for $b$, and $(4, 1, 1, 0)$ for $c$. On the other hand, no such vector exists for $d$, since the scores of $a$ and $c$ are in any case at least as high as that of $d$. In addition, there are some interesting special cases which we want to discuss here. First, we consider $\vec{\alpha}^1 = (2, 1, 1, 0)$, which lies on the edge of the winner area of $b$, but for which $b$ is nevertheless unique winner due to the non-strict inequality $\alpha_2 \geq 1$. In comparison, $\vec{\alpha}^2 = (4/3, 4/3, 1, 0)$, which also lies on the edge of the non-strict inequality $\alpha_1 \geq \alpha_2$, does not belong to any of the unique winner areas, since it lies on the tie border between $a$ and $b$. $\vec{\alpha}^3 = (8/3, 4/3, 1, 0)$, in turn, is a three-tie vector between $a$, $b$, and $c$. Considering $\mathcal{D}$-Close Scoring System with a Euclidean distance limit of 1 around $\vec{\alpha}^4 = (3, 2, 1, 0)$, we see that there are rational vectors for which $b$ or $c$ would be unique winners, but all valid integer vectors in the radius either make $a$ the unique winner or lie on a tie border.

Thus, even in this small example, we have seen that a wide variation of different combinations and properties of outcomes can arise depending on the system.

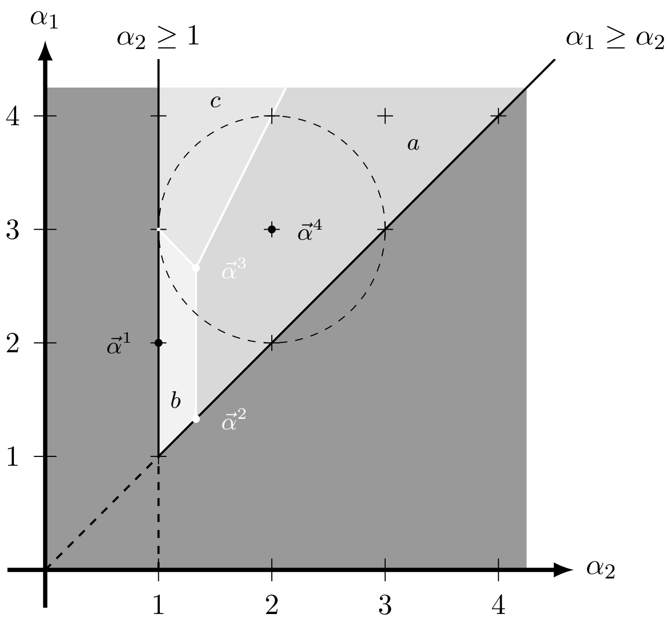


Figure 4.1: Space of scoring vectors from Example 4.1 over $(\alpha_1, \alpha_2, 1, 0)$ with respect to $\alpha_1$ and $\alpha_2$. The dark colored areas contain vectors that do not satisfy the monotonicity conditions. The other three different shaded areas are the areas which contain the vectors for which respectively $a$, $b$, or $c$ are the unique winners. The white lines between the areas are the respective, infinitely thin, tie borders. The circle around $\vec{\alpha}^4 = (3, 2, 1, 0)$, indicates a Euclidean distance of 1 from $\vec{\alpha}^4$.

We have studied the complexity, and in particular the parameterized complexity, of Scoring System Existence for several combinations of rational and integer vectors, various constraints, and parameterizations. For example, we have shown that the problem can be solved efficiently by linear programming in the unconstrained variant even for multiple profiles and rational and integer vectors. The same is true for setting an arbitrary value $\gamma$ for $\alpha_1, \dots, \alpha_{m-1}$ for rational vectors, whereas we have shown that the problem is NP-complete for integer vectors for $\alpha_{n-k} = \gamma$, even if the value $\gamma \in \mathbb{N}$ with $\gamma \geq 1$ and the position $k \geq 1$ are fixed. If we require $\alpha_k = \gamma$ for integer vectors, the problem again becomes easy to solve for fixed $\gamma$ and $k$, but if we allow $\gamma$ to be part of the input, it becomes NP-complete and W[2]-hard with respect to the parametrization by $\gamma$. For the hardness results, we have developed a general technique to construct, with respect to the monotonicity properties of the vectors, almost arbitrary linear inequalities for the scores using certain combinations of votes within a single profile. Thus, we can reduce a variety of combinatorial problems to Scoring System Existence with $\alpha_k$ or $\alpha_{n-k} = \gamma$ using their standard ILP formulations, including the NP-complete, and with respect to the size of the set also W[2]-hard, dominating set problem. With respect to the standard election parameters, the number of voters $n$ and the number of candidates $m$, respectively, the problem becomes FPT. We have studied $\mathcal{D}$-Close Scoring System for the three best known Minkowski distances, namely the Manhattan, Euclidean, and Chebyshev distance, and have shown that for rational vectors the problem can again be solved efficiently by extending the linear program mentioned earlier. For integer vectors, the distance constraint is sufficient to make the problem hard both in general and parameterized by the distance bound. In particular, it also follows from the proof that no efficient constant-factor approximation algorithm for the minimum necessary distance can exist, unless P = NP. To study the practical relevance of the problem, we implemented the previously developed integer linear programs in CPLEX and applied them with different further restrictions to the Formula 1 data in the `PrefLib` (Mattei and Walsh [2013]), which were contributed by Robert Bredereck and previously collected and used for studying the Kemeny voting rule (Betzler et al. [2014]). We have found that in many years the championships were extremely robust, in that no other scoring system exists under weak constraints where someone other than the actual champion would have won. On the other hand, in many other years even minimal changes resulting in very reasonable scoring systems would have produced different champions. Obviously, these experimental results have to be taken with a grain of salt, as this is a pure posterior analysis and changing the scoring system in practice may also lead to a change in the behavior of the participants.

Regarding related work, the problem of scoring system existence as mentioned above has been studied previously by Baumeister et al. [2011b] in the context of the possible winner problem. They have shown that the problem is NP-complete for integer vectors of the form $(\alpha_1, \dots, \alpha_{m-4}, x_1, x_2, x_3, 0)$ with $x_i = 1$ for at least one $i \in \{1, 2, 3\}$ under the assumption of succinct representation. We have significantly generalized this result by showing that it is also NP-complete for considerably less constrained vectors even without requiring the succinct representation. Parallel to our investigations and seemingly unaware of the work by Baumeister et al. [2011b], Viappiani [2018] studied the problem from a different non-complexity oriented but rather axiomatic perspective. However, he performed similar experiments, curiously enough, also on the `PrefLib` Formula 1 dataset and found that in certain years the actual champion remains a winner independent of the actual scoring system used, just as we did. Uncertainty about the voting rule was also studied by Elkind and Erdélyi [2012], who extended manipulation and coalitional manipulation to include a set of voting rules rather than a single one, and the manipulator(s) try to manipulate in such a way that their candidate would win with respect to each of the rules. They also present an interesting observation, referred to a private communication with Jérôme Lang, that states that for a given profile a distinguished candidate can only be the unique winner with respect to all integer scoring rules if and only if the candidate is also the unique winner with respect to $k$-approval for each $k \in \{1, \dots, m-1\}$. The latter condition is therefore an easy to verify criterion to show that no scoring system exists under any constraint that makes another candidate the unique winner, and if it fails, we already have a scoring rule by the respective $k$-approval that proves the dependence of the candidate's victory on the scoring rule in place.

While we have focused here so far, and also in the publication, predominantly on related work in the context of computational complexity, there is a significant body of work that has studied the automated data-based design of scoring rules, and voting rules in general, using different approaches. We stress the 'automated' here because the manual design of voting rules based on axioms, assumptions, and/or intuitions already has a long and rich history, as mentioned in the introduction. All of the following approaches basically pursue the same objective as we do here: finding and/or optimizing a voting rule with respect to certain criteria. The main differences are the criteria, namely whether we try to reproduce the given outcome of certain profiles or, e.g., to maximize the underlying utility functions of the voters, and the actual way of finding and optimizing the rule. Procaccia et al. [2009] studied the design of scoring rules with respect to the concept of probably approximately correct (PAC) learning as introduced by Valiant [1984], which forms the theoretical framework for the concept of machine learning. In their scenario, there is a consultable agent that serves as a black box oracle and determines a winner for each given profile. The goal is to find a scoring rule with as few queries to the agent as possible, which determines the same winner with high probability for a given distribution. In the case that the agent itself generates its answers according to an underlying scoring rule, they show that an efficient learning algorithm exists which generates a scoring rule that is sufficiently accurate, making the problem itself efficiently PAC-learnable. Thus, the objective here is similar to ours, with the difference that in our problem, we are already given the profiles together with the outcome as a list, instead of the consultable agent. Very recently, Caragiannis and Fehrs [2021] have applied this approach to a generalization of scoring rules to multi-winner elections, the so-called approval-based committee scoring rules. Referring back to single-winner elections, a similar approach, but with underlying utility functions of the voters generating the ordinal preferences and the goal to find a scoring rule that minimizes the distortion, i.e., the relative loss in utility through using the respective voting rule instead of maximizing the utilities, in the worst case, on average, or with respect to a set of learning samples has been studied by Boutilier et al. [2015]. Later, Caragiannis et al. [2019] studied the problem of finding and optimizing scoring rules that reproduce an underlying partially known truthful ranking in a scenario where each voter submits an ordinal preference only over a small subset of the candidates, which may differ from voter to voter. On the practical side, Burka et al. [2016] trained neural networks to predict winners using random profiles with certain properties. In their comparison most networks tended to predict the winners according to the Borda rule. However, as they correctly point out in a recent revision (Burka et al. [2021]), neural networks, even with only one hidden layer, are able to approximate any voting rule pretty well under the assumption of a sufficient width of the layer and the size of the training set, according to the universal approximation theorems regarding neural networks (see, e.g., Cybenko [1989]). Nevertheless, for reasonable sizes of the network and the training set the neural networks under consideration tend to predict according to Borda. Regarding multi-winner rules, a popular way of transferring single-winner scoring rules to multi-winner elections is to apply ordered weighted average aggregation (OWA) introduced by Yager [1988]. Here, each voter assigns a score to each possible committee based on the scalar product of the OWA vector and the respective scores of the candidates in the committee with respect to a scoring rule sorted by the voter's preference. Thus, OWA-based multi-winner rules allow us to model a diminishing return of utility that a voter gets from the candidates in the committee, making it possible to scale between multi-winner rules like $k$-Borda, which emphasize individual strength of candidates in the committee, and rules like Chamberlin-Courant, which emphasize diversity and representation of voters through the candidates in the committee. Recently, Faliszewski et al. [2018, 2022] studied the design of such OWA-based rules by considering so-called utopic distributions of winners, the distribution of candidates in the winning committees over the space from which the votes and candidates are drawn according to the Euclidean model, for which one can infer the objectives of the rule, such as individual excellence or diversity. Their approach is to fix either the OWA vector or the scoring rule and to optimize the other vector regarding the utopic distribution using a randomized search algorithm. Interestingly, some of the resulting vectors actually corresponded to existing rules with similar objectives, which supports the relevance of those very rules, while in other cases new rules were found.

## 4.2 Publications

This work was published as:

> D. Baumeister and T. Hogrebe. How Hard Is the Manipulative Design of Scoring Systems? In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 74–80, 2019a.

A preliminary version of this publication appeared as an extended abstract in the proceedings of the *18th International Conference on Autonomous Agents and Multiagent Systems*:

> D. Baumeister and T. Hogrebe. Manipulative Design of Scoring Systems (Extended Abstract). In *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems*, pages 1814–1816. IFAAMAS, 2019b.

## Personal Contribution

The writing and development of the model was done jointly with Dorothea Baumeister. The technical results were contributed by me.

# Chapter 5

## Prediction and Probabilistic Robustness of Elections

This chapter deals with the function complexity of calculating the winning probability of candidates in scoring rule and approval elections regarding different vote distributions.

### 5.1 Summary

When elections are discussed in the real-world, it is usually about two aspects: how an upcoming election will probably turn out and how reliable are the results for a conducted election. Hence, in the following we will study these two problems, namely the problem of predicting the outcome of an election and the problem of determining the robustness of the outcome of an election.

However, compared to the possible winner problem by Konczak and Lang [2005] and the robustness problem by Shiryaev et al. [2013] considered in Section 2.2 and Chapter 3, we want a more comprehensive and richer output, which is oriented to more than one possible completion or modification of the election profile and allows for a more differentiating comparison of the candidates. Therefore, we adopt the framework of election evaluation introduced by Conitzer et al. [2007] and Hazon et al. [2012] in which the objective is to determine the winning probabilities of candidates with respect to given distributions over possible profiles. Formally the *evaluation problem* is given as follows.

| $\mathcal{E}$-Evaluation | |
|---|---|
| **Given:** | A set of candidates $C$, a profile distribution $\mathcal{P}$ over $C$, and a distinguished candidate $p \in C$. |
| **Question:** | What is the probability $\Phi$ that $p$ is a winner of the election with respect to $\mathcal{E}$ assuming that the profiles are distributed according to $\mathcal{P}$? |

Note that the problem as we state it here is quite generic and the motivation and complexity depends heavily on the profile distribution and its encoding. For example, if one chooses a distribution that is concentrated on one profile the problem is equivalent to the winner determination problem and thus easy to solve for many voting rules. We focus on the following three distributions. Note that all three of them are composed of independent distributions over the possible ordinal/approval votes for each voter. Thus, we assume that in our scenarios voters independently determine their vote. However, other scenarios are conceivable in which due to group dynamics and large-scale influence the votes are correlated.

*EDM.* In the model which we refer to as the *explicit distribution model* (EDM), the votes with positive probability for each voter are explicitly given as a list paired with their respective probabilities as rational numbers. It was studied by Conitzer et al. [2007] and it is also the model on which the problem was defined by Hazon et al. [2012].

*PPIC.* In the *partial profile impartial culture model* (PPIC), we assume that we are given a partial profile and consider a uniform distribution over the possible completions. Thus, the problem in this case becomes the normalized counting variant of the possible winner problem.

*Mallows.* In the *Mallows noise model* (Mallows), introduced by Mallows [1957] for ordinal votes, we assume that every additional swap of two adjacent candidates in an ordinal preference or the flipping of an approval entry in an approval vote with respect to a given central profile lowers the probability by a factor $\varphi \in \mathbb{Q} \cap (0, 1)$, referred to as the *dispersion parameter*.

For distributions with independent voters, the EDM is the most general with respect to expressiveness, since all such distributions can be expressed by explicitly listing the votes with positive probability. However, from a computational point of view, the transformation of distributions like PPIC or Mallows to EDM is not feasible, since it may require $\mathcal{O}(2^m)$ votes to be listed in the case of approval votes and even $\mathcal{O}(m!)$ votes to be listed in the case of ordinal votes for each voter.

The presented models can be used to capture a wide range of scenarios through the evaluation problem. For example, using PPIC, one can perform *election predictions* for upcoming elections using partial data collected through polls or the (automatic) aggregation of social media data. However, it is assumed here that each possible completion for a partial vote has the same probability. While this assumption may be reasonable depending on the scenario or may even be necessary in practice due to missing additional data, it may also be the case that certain patterns or biases are known. In that case, the EDM can be used to model such assumptions, but possibly at the expense of the compactness of the representation. Another highly relevant scenario that can be modeled under PPIC is the *winner determination using partial data*. The reasons for partial election data can be diverse. Based on the circumstances, it may simply be too difficult or costly, or even impossible, to collect complete election data. It may also be that the number of candidates standing for election or the way in which votes are collected changes during the election, and thus some of the votes are not complete according to the final record. Another recurring problem, especially in elections with paper ballots, is the loss of election data during collection, counting, or transmission. However, if a winner has to be determined in these scenarios, the usual first approach is to check whether more than one candidate can be considered as the winner at all. If not, the winner is already determined. This is the approach of the classical possible winner problem. However, if more than one candidate is possible, we face a problem. How to decide between the candidates? Especially if each of the candidates claims victory for itself. In these cases, determining the probability of victory can be of immense benefit. In the worst case, all candidates have the same chance of winning and we know that a winner determination via tie-breaking or re-election will be necessary. In the best case, one of the candidates has a winning probability close to one, whereby he or she is probably the rightful winner. However, even if the election data is complete, we cannot be sure that it is flawless. For example, in many situations we have to assume that voters have made mistakes when casting their votes, that there have been small-scale manipulations, or that the data have been corrupted during collection or transmission. In these situations, it is natural to ask how justified a candidate's victory is under such uncertainties and how close other candidates are to winning. This is also the motivation behind the margin-of-victory problem and the robustness problem which we discussed in Section 2.2.3. However, the answers to these problems are based only on single alternative profiles and, thus, may miss the bigger picture. Using Mallows and the evaluation problem, we obtain a probability that incorporates all possible profiles based on the dispersion parameter, which is adjustable to the degree of uncertainty in the data. Thus, in contrast to the previous presented robustness measures we refer to this approach as the *probabilistic robustness*.

However, these advantages we praise above come with immense computational cost in some cases. We have shown that Evaluation assuming Mallows is #P-hard for $k$-approval and $k$-veto respectively for fixed $k \geq 1$, thus including plurality and veto, Borda, and $(2, 1, \ldots, 1, 0)$. For this, we have shown that the evaluation problem for Mallows and the counting variant of the unweighted swap bribery problem (see Chapter 3) are equivalent under polynomial-time Turing reduction. By leveraging a reduction by Bachrach et al. [2010], which studied the unnormalized counting variant of the possible winner problem, we were able to show that the problem is #P-hard for PPIC and pure scoring rules even if each preference in the profile has at most two possible completions. Since in this specific case PPIC instances can be efficiently transformed into equivalent EDM instances, the #P-hardness for pure scoring rules also holds for EDM, thereby completing the results by Hazon et al. [2012] regarding pure scoring rules. Moreover, we have also shown that for $k$-AV with $k \geq 1$ the problem is #P-hard for PPIC and EDM, and for AV under EDM. On the positive side, for AV the problem is solvable in polynomial time for both PPIC and Mallows using dynamic programming.

Due to the predominant computational hardness, we examined the parameterized complexity with respect to constant numbers of candidates or voters in the next step. For a constant number of candidates, PPIC and Mallows instances can be efficiently transformed into equivalent EDM instances, allowing us to solve the problem efficiently in that case using the polynomial-time algorithm by Hazon et al. [2012]. For a constant number of voters, we could show for almost all cases that the problem can be solved efficiently using different approaches. However, we found that for PPIC the problem is #P-hard in the non-unique winner case for all pure scoring rules even for a single voter. However, this result does not hold for the unique winner case, since in this case the problem for certain scoring rules can be trivial for a low number of voters, making the complexity in this case much more diverse. For example, the problem in that case is trivial for 2-approval for one voter, but #P-hard for two. For veto, on the other hand, the problem is never #P-hard for a constant number of voters, since it becomes trivial as soon as the number of candidates exceeds the number of voters by more than one.

Apart from the case just mentioned, where the complexity may differ, all other results apply to both the non-unique and the unique winner case. However, both models have a disadvantage in terms of probability determination: the winning probabilities of the candidates do not necessarily add up to one. In the non-unique winner case, profiles can be counted several times, which is why the sum can be higher than one. In the unique winner case, profiles may not be counted at all, which is why the sum may be less than one. To solve this problem, we have also shown that all our results for the non-unique case also hold for uniform random tie-breaking and lexicographic tie-breaking, for which the winning probabilities of the candidates form a valid probability distribution.

Conitzer et al. [2007] originally defined the evaluation problem not as a weighted counting problem but as a decision problem in which a rational number $r$ is given and the question is whether the probability of victory is greater than $r$. In fact, the evaluation problem originated as the verification problem for the manipulation problem with uncertain votes, thus, to check whether a manipulation is successful with a certain minimum probability. They show several NP-hardness results for manipulation and evaluation with $r = 0$, but also observe that the manipulation problem for arbitrary $r$ is not necessarily in NP. We support this observation by showing that the decision problem and the weighted counting problem are equivalent under polynomial-time Turing reduction for the models considered. Thus, the #P-hardness of the evaluation problem as the verification problem makes the membership of the manipulation problem in NP indeed very unlikely in the respective cases.

Very recently, Boehmer et al. [2021a] considered a similar approach to election robustness by studying the counting variants of the swap bribery and shift bribery problems. In comparison to the Mallows model, however, not all profiles are weighted according to their distance to the original profile, but all profiles up to a given distance limit are weighted equally.

Problems corresponding to the evaluation problem have also been studied in many related areas. For example, Fazzinga et al. [2015] studied the problem in the context of abstract argumentation, Aziz et al. [2019] in the context of resource allocation, and Aziz et al. [2020] in the context of stable matching. In sports, the evaluation problem was initially studied by Mattei et al. [2015], and recently also by us. We will discuss the evaluation problem in sports in much more detail in Chapter 6.

## 5.2 Publications

This work was submitted by invitation to SNCS for the EUMAS 2021 special issue:

> D. Baumeister and T. Hogrebe. On the Complexity of Predicting Election Outcomes and Estimating Their Robustness. *SN Computer Science*, Submitted.

Preliminary versions of this work appeared in the proceedings of the *18th European Conference on Multiagent Systems* and as an extended abstract in the proceedings of the *19th International Conference on Autonomous Agents and Multiagent Systems*:

> D. Baumeister and T. Hogrebe. On the Complexity of Predicting Election Outcomes and Estimating Their Robustness. In *Proceedings of the 18th European Conference on Multiagent Systems*, pages 228–244. Springer, 2021b

and

> D. Baumeister and T. Hogrebe. Complexity of Election Evaluation and Probabilistic Robustness. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems*, pages 1771–1773, 2020.

## Personal Contribution

The writing and development of the model was done jointly with Dorothea Baumeister. The technical results were contributed by me.

# Chapter 6

## Predicting Round-Robin Tournaments

This chapter deals with the worst-case and average-case function complexity of calculating the winning probabilities of teams in round-robin tournaments.

## 6.1 Summary

As for elections in the previous chapter, one of the most discussed real-world problems for sports tournaments is the prediction of their outcomes. However, many sports tournaments, and in particular the round-robin tournaments studied here, have a crucial difference compared to elections. While most elections are held in a short period of time and counting does not begin until all voters have cast their ballots, most tournaments span a significant period of time with respect to a given schedule. Thus, for tournaments, it is particularly realistic and interesting to consider the case where we are at a certain point in the course of a tournament where some matches and matchdays have already been played and others are still open.

To capture the state of a given score-based pair-wise tournament $\mathcal{T} = (T, M)$ over a set of outcomes $O$ we use a so-called *outcome probability profile* $\rho = (\rho_m)_{m \in M}$, where $\rho_m : O \to \mathbb{Q}$ denotes the probability distribution over the outcomes $O$ for match $m$. Analogously to the evaluation problem for elections in Chapter 5, we can now define the evaluation problem for tournaments. Assume we are given a fixed set of outcomes $O$.

| $O$-Evaluation | |
|---|---|
| **Given:** | A score-based pair-wise tournament $\mathcal{T} = (T, M)$, an outcome probability profile $\rho$ for $\mathcal{T}$ with respect to $O$, and a distinguished team $p \in T$. |
| **Question:** | What is the probability that $p$ ends up as the unique winner of the tournament with respect to $\rho$? |

As mentioned above, our main focus here is on the consideration of scheduled round-robin tournaments, as they usually occur in the real world. We refer to $O$-Evaluation restricted to round-robin tournaments given together with an optimal schedule as $O$-RRTS-Evaluation. Something we have not explained yet, however, is how we model the scenario of observing the tournament at a certain point in time, when certain matchdays have already been played and others still remain. For this we make the following definition. We refer to a match $m$ as open if $\rho_m$ assigns a probability greater than 0 to more than one outcome and refer to a matchday as open if at least one of the assigned matches is open. If we now assume that we are in a tournament at a certain point in time, a match $m$ that lies in the past that has already been played is not allowed to be open and must have a unique outcome with probability 1. Thus, in the input of the evaluation problem, we do not explicitly distinguish the already played matches from the remaining matches as in the definition of the elimination problem in Section 2.3, but rather encode this in a compact way in the outcome probability profile.

We will now illustrate the problem with an example.

**Example 6.1.** We consider the round-robin tournament $\mathcal{T} = (T, M)$ with $T = \{t_1, t_2, t_3, t_4, t_5, t_6\}$ and $O = \{(3,0), (1,1), (0,3)\}$ from Example 2.9. A possible outcome probability profile over this tournament might look as follows.

| Matchday 1 | $(3,0)$ | $(1,1)$ | $(0,3)$ |
|---|---|---|---|
| $m_1 : (t_1, t_6)$ | 0 | 1 | 0 |
| $m_2 : (t_2, t_5)$ | 1 | 0 | 0 |
| $m_3 : (t_3, t_4)$ | 0 | 1 | 0 |

| Matchday 2 | $(3,0)$ | $(1,1)$ | $(0,3)$ |
|---|---|---|---|
| $m_4 : (t_1, t_5)$ | 1 | 0 | 0 |
| $m_5 : (t_2, t_3)$ | 1 | 0 | 0 |
| $m_6 : (t_4, t_6)$ | 0 | 0 | 1 |

| Matchday 3 | $(3,0)$ | $(1,1)$ | $(0,3)$ |
|---|---|---|---|
| $m_7 : (t_1, t_4)$ | 1 | 0 | 0 |
| $m_8 : (t_2, t_6)$ | 1 | 0 | 0 |
| $m_9 : (t_3, t_5)$ | 0 | 0 | 1 |

| Matchday 4 | $(3,0)$ | $(1,1)$ | $(0,3)$ |
|---|---|---|---|
| $m_{10} : (t_1, t_3)$ | 0.7 | 0.2 | 0.1 |
| $m_{11} : (t_2, t_4)$ | 0.8 | 0.1 | 0.1 |
| $m_{12} : (t_5, t_6)$ | 0.4 | 0.2 | 0.4 |

| Matchday 5 | $(3,0)$ | $(1,1)$ | $(0,3)$ |
|---|---|---|---|
| $m_{13} : (t_1, t_2)$ | 0.3 | 0.4 | 0.3 |
| $m_{14} : (t_3, t_6)$ | 0.5 | 0.3 | 0.2 |
| $m_{15} : (t_4, t_5)$ | 0.4 | 0.3 | 0.3 |

As described above, matches that have already been played are assigned their results in the outcome probability profile by setting the probability for the respective outcome to 1.

Now we focus on $t_1$. As shown in Example 2.9, $t_1$ with 7 points after the third matchday can become the unique winner of the tournament, despite $t_2$ leading with 9 points, for example if $t_1$ wins its two remaining matches, regardless of the outcomes of the other matches. For the determination of the probability that $t_1$ will end up as the unique winner, however, we cannot make it so simple, since we essentially have to consider every possible combination of outcomes of the remaining matches:

- As said, $t_1$ is guaranteed to be the unique winner if it wins both of its remaining matches, regardless of the other outcomes. The probability for this is $0.7 \cdot 0.3 = 0.21$.
- Second, $t_1$ is guaranteed to be the unique winner if $t_1$ wins against $t_3$ and draws against $t_2$, whereby $t_2$ has to draw or lose against $t_4$, regardless of the other outcomes. The probability for this is $0.7 \cdot 0.4 \cdot (0.1 + 0.1) = 0.056$.
- Third, $t_1$ is guaranteed to be the unique winner if $t_1$ draws against $t_3$ and wins against $t_2$, again, whereby $t_2$ has to draw or lose against $t_4$, regardless of the other outcomes. The probability for this is $0.2 \cdot 0.3 \cdot (0.1 + 0.1) = 0.012$.

Thus, the probability for $t_1$ to end up as the unique winner is $0.21 + 0.056 + 0.012 = 0.278$.

Note that in the above example we took several shortcuts in determining the probability by first considering matches whose outcomes may render the outcomes of other matches irrelevant. This is also the key idea of our FPT algorithm which we will present later.

As in the previous Chapter 5 regarding elections, where the evaluation problem assuming PPIC may be seen as a weighted counting variant of the possible winner problem, here the evaluation problem can be seen as a weighted counting variant of the possible winner problem for tournaments, the elimination problem. The reduction of the elimination problem to the evaluation problem is thus simple and can be done in polynomial time. All matches played get their unique outcomes in the outcome probability profile by setting the respective probability to 1, for all remaining matches we set the probabilities according to a uniform distribution over the outcomes. In order to answer the elimination problem, we only need to check if the probability calculated by the evaluation problem is greater than 0. Thus, the worst-case efficiency results, and also the average-case efficiency results presented here, carry over from the evaluation problem to the elimination problem. In the other direction, the NP-hardness of the elimination problem is also passed on to the evaluation problem. However, we will see later that the hardness of the evaluation problem usually lies far above NP-hardness and in some cases it is computationally hard even if the elimination problem can be solved efficiently in the respective cases.

The computational complexity of the evaluation problem regarding round-robin tournaments has only been studied by Mattei et al. [2015] and by Saarinen et al. [2015] before us. Saarinen et al. [2015] have shown that the evaluation problem for round-robin tournaments with $O = \{(1,0),(0,1)\}$, also referred to as Copeland tournaments in the context of elections, is indeed #P-hard even if all outcome probabilities are from $\{0, 1/2, 1\}$. This result is in contrast to the result of Schwartz [1966] regarding the elimination problem, which we discussed in Section 2.3 and which states that the elimination problem for $O = \{(1,0),(0,1)\}$ is solvable in polynomial time. However, Saarinen et al. [2015] do not take into account the existence of a schedule in their reduction from the problem of counting Eulerian orientations, for which Mihail and Winkler [1996] showed #P-completeness. This manifests itself in particular in the fact that teams can have widely different numbers of open matches.

Thus, we were interested in understanding how the computational complexity of the problem behaves under the conditions described above, and in particular how the complexity depends on the number of open matchdays. We were able to show that the problem is still #P-hard for all symmetric sets of outcomes, even when restricted to scheduled round-robin tournaments where only a fixed number of at least three matchdays remains open and all outcome probabilities are from $\{0, 1/2, 1\}$. On the other hand, we showed that the problem can be solved in polynomial time if there are at most two open matchdays This completes our dichotomy result regarding the complexity of the problem with respect to the number of remaining matchdays. In fact, the efficiency result for at most two open matchdays is only a corollary of a more general efficiency result we proved. It states that the evaluation problem for score-based pair-wise tournaments, and thus in particular for round-robin tournaments, is fixed parameter tractable with respect to the so-called maximum fixing number. Loosely speaking, this parameter scales with the number and density of matches between teams that can be dangerous for our distinguished team in terms of becoming the champion.

Something that made the FPT parameter of the algorithm so intriguing to us, besides the theoretical implications, was the impression that it might often be quite small for real-world instances. Therefore, we collected data from 140 seasons from the major European football leagues and determined the average maximum execution time of the algorithm across all teams as a function of the number of open matchdays. While the average execution time increased significantly with the number of open matchdays, it was still below 60 seconds for 12 open matchdays, which corresponds to about one-third of the season for the seasons considered. In comparison, the trivial brute force approach would have to consider between $3^{12\cdot(16/2)} \approx 6.363 \cdot 10^{45}$ and $3^{12\cdot(20/2)} \approx 1.797 \cdot 10^{57}$ combinations, which is already not feasible in practice. It should be mentioned here that while there are several different approaches in the literature for solving the elimination problem such as ILP formulations, FPT algorithms, and in some cases even polynomial-time algorithms (see Section 2.3), the algorithm we have developed here is, to the best of our knowledge, the first approach for actually solving the evaluation problem.

However, there was not enough real-world data to examine the behavior of the execution time with respect to the number of participating teams. Therefore, we implemented three different models by Ryvkin and Ortmann [2008] to generate hundreds of seasons for different numbers of teams. Interestingly, and in contrast to the computational hardness we had previously shown, the execution time for the three models dropped again by several orders of magnitude after an initial spike and then increased only very slowly afterwards. These empirical observations led us to investigate the actual average-case complexity of the evaluation problem for the worst-case computational hard cases using our algorithm. And indeed, we were able to confirm our observations by proving that the expected maximum execution time of the algorithm across all teams is polynomially bounded for a distribution that seems to dominate the distribution of real-world instances with respect to the complexity of the algorithm. As previously mentioned, those average-case efficiency results also carry over to the elimination problem, for which again no average-case results were known before.

A general criticism that can be made regarding the model used for evaluation here is the independence of the outcome probabilities between matches. In real-world tournaments, it is often observed that the performance of teams depends on the results in previous matches. This is not only due to perceived trends and the resulting motivation or demotivation, but especially due to the fact that at the end of the season a team may have already reached its goal or may not be able to reach it anymore and thus may not perform to its full potential in the following matches. Of course, the other direction is also often observed, namely that a team that is relying on winning a certain match to reach its goal due to previous results performs significantly better than expected. The modeling of such mechanics alone can be an extremely complex step, which in many cases is probably not possible with polynomial effort and takes the focus away from the computational complexity of the actual problem. Note, however, that the case of independent outcome probabilities is a special case of dependent outcome probabilities and therefore our hardness result also applies to that very case.

Another possibility for future work is to investigate other tournament formats such as the Swiss system, which is often used in chess. This is particularly interesting because, unlike round-robin tournaments, the set of remaining matches is determined by the outcomes of the previous matches.

## 6.2 Publications

This work was published as:

> D. Baumeister and T. Hogrebe. Complexity of Scheduling and Predicting Round-Robin Tournaments. In *Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems*, pages 178–186. IFAAMAS, 2021a

and

> D. Baumeister and T. Hogrebe. On the Average-Case Complexity of Predicting Round-Robin Tournaments (Extended Abstract). In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*. IFAAMAS, 2022.

## Personal Contribution

For both publications, the writing and development of the model was done jointly with Dorothea Baumeister. The technical results were contributed by me.

# Chapter 7

## Smoothed Complexity Analysis

This chapter deals with the possibilities of applying the concept of smoothed analysis proposed by Spielman and Teng [2004, 2009] in the context of computational social choice.

## 7.1 Summary

As discussed in Section 2.1.4, and shown in Chapter 6 regarding the prediction of round-robin tournaments, average-case complexity can explain why in practice many problems can be solved much faster than their worst-case hardness suggests. However, often the main difficulty is not to prove that a problem can be solved efficiently with respect to a certain distribution in the average-case, but to find a meaningful distribution in the first place, which approximates the real-world distribution or at least, as in the previous chapter, seems to dominate the real-world distribution with respect to the complexity of the problem. To overcome this problem Spielman and Teng [2004, 2009] introduced the smoothed complexity analysis. The smoothed complexity analysis of an algorithm represents a middle ground of worst-case and average-case complexity by asking what the expected worst-case running time of the algorithm is under the assumption that the instances are randomly perturbed. This assumption is well founded if one assumes that the instances originate from real-world sources that are subject to natural noise, such as data from sensors, including photography, sound recordings, and other physical measurements, sensitive personal data that have been slightly disturbed to protect privacy, or in general, data that have been interfered with in collection, transmission, or storage.

We now introduce the formal definitions of *smoothed complexity analysis*. We start with the original definitions which Spielman and Teng [2004, 2009] introduced to explain the practical efficiency of the Simplex algorithm beside its exponential worst-case running time. The *smoothed complexity* of an algorithm $A$ over $[-1, 1]^n$ with respect to a Gaussian perturbation with standard deviation $\sigma$ is given by

$$\text{Smoothed}_A^\sigma(n) = \max_{\hat{x} \in [-1,1]^n} \text{E}\left[T_A(\hat{x} + \delta)\right]$$

where $\delta \in \mathbb{R}^n$ is a random vector in which each entry is sampled from the univariate Gaussian distribution with standard deviation $\sigma$. Thus, instead of measuring the worst-case running time over all instances, we measure the expected worst-cast running time assuming a perturbation of all instances. Note that for $\sigma = 0$ we retrieve the worst-case running time, while for large $\sigma$ we approach an average-case analysis. Note, however, that the focus here is particularly on small perturbations around the central instance, which is also reflected in the following definition. An algorithm $A$ has *polynomial smoothed complexity* if there exist positive constants $n_0$, $\sigma_0$, $c$, $k_1$, and $k_2$ so that for all $n \geq n_0$ and $0 < \sigma \leq \sigma_0$ it holds that

$$\text{Smoothed}_A^\sigma(n) \leq c \cdot (1/\sigma)^{k_1} \cdot n^{k_2}.$$

Thus, here we require that the algorithm has polynomially bounded smoothed complexity not only as a function of the number of variables $n$, but also with respect to $1/\sigma$. As mentioned before, these definitions represent a middle ground between worst-case and average-case analysis, and as for the definition of average-case polynomial time, see Section 2.1.4, there is also a weaker but more robust definition here. An algorithm $A$ has *probably polynomial smoothed complexity* if there exist positive constants $n_0$, $\sigma_0$, $c$, and $\varepsilon$ so that for all $n \geq n_0$ and $0 < \sigma \leq \sigma_0$ it holds that

$$\max_{\hat{x} \in [-1,1]^n} \mathrm{E}\left[(T_A(\hat{x} + \delta))^\varepsilon\right] \leq c \cdot (1/\sigma) \cdot n$$

where $\delta \in \mathbb{R}^n$ is a random vector in which each entry is sampled from the univariate Gaussian distribution with standard deviation $\sigma$.

Smoothed complexity analysis has been successfully applied to many algorithms to explain the efficient solvability of the respective problems in practice. For detailed overviews of the various applications and aspects, we refer to the surveys by Spielman and Teng [2009] and Manthey and Röglin [2011]. Naturally, the best-known and most widely recognized application of smoothed analysis is its original application, the explanation of the efficiency of the Simplex algorithm for linear programming in practice by Spielman and Teng [2004] despite its exponential worst-case runtime. Another algorithm frequently used in practice is the $k$-means algorithm for the clustering of data points, one of the main tasks in unsupervised machine learning. Arthur et al. [2011] explained the practical efficiency of $k$-means by proving its polynomial smoothed complexity, despite the worst-case exponential runtime shown by Vattani [2011]. Other examples include the results of Banderier et al. [2003] on sorting and the shortest path problem, of Beier and Vöcking [2004] on the algorithm by Nemhauser and Ullmann [1969] for the Knapsack problem, of Englert et al. [2014] on the traveling salesman problem, and of Etscheid and Röglin [2017] on the maximum-cut problem. In addition to time complexity, smoothed analysis can also be used to study other performance measures of an algorithm or mechanism. Spielman and Teng [2009] give some examples, such as the approximation ratio of an approximation algorithm or the convergence rate of a best-response dynamic.

Here we were interested in exploring the possibilities of applying the concept of smoothed analysis to both computational and axiomatic examinations in the area of social choice. The main reasons for the attractiveness of smoothed analysis in the field of computational social choice is the rich amount of computational hardness results for various problems and the always accompanying sentiment that many of these results are quite intriguing from a theoretical perspective, but probably not relevant in practice. However, proving the latter fails in most cases in finding a suitable distribution of instances. Fortunately, preferences in the real-world are subject to natural fluctuations and uncertainties, which satisfies the basic premise of smoothed analysis. As a possible starting point, we suggested to study the computational complexity of classical election problems such as winner determination, manipulation and bribery (see Section 2.2), assuming that preferences are subject to perturbation according to the Mallows model (see Chapter 5). We pointed out that it might be reasonable to consider the definitions of smoothed analysis that have been adapted to discrete problems, such as those of Beier and Vöcking [2004] and Bläser and Manthey [2012]. In addition, we suggested to investigate the relevance of other phenomena such as voting paradoxes and ties using the smoothed framework.

Subsequent to our proposal paper, a series of papers appeared that successfully applied the concept of smoothed analysis to the area of social choice. Xia [2020] studied the smoothed likelihood of the Condorcet paradox and the satisfiability of combinations of certain voting rule axioms, which, in theory, cannot be combined due to known impossibility results. As suggested by us, Xia [2021] also examined the likelihood of ties in elections in the smoothed framework with respect to many different voting rules. Very recently, Flanigan et al. [2022] proposed an axiomatic model that can be positioned between our more specific proposal and the very general model of Xia [2020]. The key difference between their model and that of Xia [2020] is that they introduce a general distinct dispersion parameter with very mild restrictions that control the strength of the noise. However, Flanigan et al. [2022], as well as Xia [2020], consider the Mallows model as a central application of their models.

Xia and Zheng [2021] studied the smoothed complexity of computing rankings according to the Kemeny and Slater voting rule. They show that if one considers the model of Bläser and Manthey [2012] with a fixed dispersion parameter, which bounds the highest probability an instance can have, one can solve the two problems considered, but by the argumentation also many others, in smoothed polynomial time. However, fixing the dispersion parameter is explicitly not allowed in the model of Bläser and Manthey [2012] and leads the definition ad absurdum: fixing the dispersion parameter allows an instance, and for the usual perturbation distributions, the central worst-case instance, to have a fixed minimum probability independent of the input length. Thus, in this case, a reasonable definition of smoothed polynomial time for discrete problems must admit a super polynomial running time assuming that $\mathrm{P} \neq \mathrm{NP}$ holds. However, they use this as an argument that the definition is inappropriate for the study of problems in the field of computational social choice. Consequently, they observed that assuming the definitions of Spielman and Teng [2004], which were not designed for discrete problems, the problems have a smoothed polynomial time for perturbation models only if $\mathrm{NP} = \mathrm{RP}$ holds, which is generally not assumed. Note that in comparison to the definitions of Spielman and Teng [2009], they do not allow for a polynomial dependence of the smoothed running time on the strength of the perturbation.

## 7.2 Publications

This work was published as:

> J. Rothe, D. Baumeister, and T. Hogrebe. Towards Reality: Smoothed Analysis in Computational Social Choice. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems*, pages 1691–1695, 2020.

### Personal Contribution

The writing and development of the models was done jointly with Dorothea Baumeister and Jörg Rothe.

# Chapter 8

## Conclusions

In this chapter we summarize the thesis, discuss the results, and provide directions for possible future work.

### 8.1 Results

We will start our conclusions by summarizing our studies in the previous chapters and highlighting our key results.

We started in Chapter 3 by studying the problem of constructive and destructive priced bribery in scoring rule elections with respect to the weighted swap distance and the weighted footrule distance and their variants with different levels in expressiveness, namely the unweighted variant, the element-weighted variant, and the fully weighted variant. For the considered scoring rules, we were able to determine the complexity in almost all cases with respect to the classical decision complexity, namely membership in P versus NP-completeness. For the class of pure scoring rules, we identified the decisive factors for the complexity in the case of the constructive bribery and the element-weighted variants of the two distances and were thus able to establish dichotomy results. We showed that the complexity of the problem depends strongly on the combinations of prerequisites in each case, gave reductions that show NP-hardness even in highly restricted special cases, and answered open questions from the literature.

In Chapter 4 we turned to the problem of designing scoring rules and scoring systems for elections and competitions in such a way that they guarantee the victory of a particular candidate. We were able to strengthen the only result for the problem so far, by Baumeister et al. [2011b], by showing that the problem is NP-complete as soon as we specify any nonzero value at any posterior position in the integer scoring vector, not only if one of the last three values prior to the final zero is one, and even without the assumption of succinct representation. In addition, we examined the extension of the problem in which we assume that a scoring system is already in place, which we want to change as little as possible with respect to the Manhattan, Euclidean or Chebyshev distance. For these cases we again proved NP-completeness for integer vectors. In addition to the classical decision complexity, we investigated the parameterized complexity of the problems with respect to various natural parameters such as the number of candidates and voters, the fixed value in the scoring vector, and the distance limit, for which we showed either membership in FPT or W[2]-hardness. Finally, we showed by experiments using Formula 1 results that the problem is relevant in practice.

In Chapter 5 we studied the problem of determining the winning probability of candidates in elections with probabilistically distributed votes, the so-called evaluation problem. For almost all combinations of the two approval voting rules and the considered scoring rules, preference distribution models, winning cases, and parametrization of the number of candidates and voters, we determined the complexity, in terms of membership in FP versus #P-hardness. Moreover, we extended all results to random and lexicographic tie-breaking and showed the polynomial-time equivalence of the function and decision variant of the problem. In addition, we showed dichotomous results for the class of scoring rules assuming PPIC or EDM as the distribution model and showed the polynomial-time equivalence of the evaluation problem assuming Mallows and the counting variant of the unweighted swap bribery problem, which we studied in Chapter 3.

In Chapter 6, we then moved from elections to sports tournaments, more precisely round-robin tournaments, and studied the complexity of the evaluation problem for these as well under the assumption that the probabilities for the outcomes of the remaining matches are given. We showed that the problem is #P-hard as soon as at least three matchdays remain and is in FP otherwise. We thus strengthened the existing result of Saarinen et al. [2015] in that the hardness also holds if one assumes the existence of a schedule and all remaining matches are not spread over the whole tournament, but compactly on the last matchdays. To better understand the complexity of the problem in practice, we then considered the parameterized complexity with respect to a specific parameter, the maximum fixing number, which scales with the density of matches between the top teams and developed an FPT algorithm with respect to this parameter. As a next step, we implemented the algorithm and examined its running time using real data and synthetic data. We found that the running time does not grow as expected for a #P-hard problem under the usual assumptions, which is why we subsequently examined the average-case complexity of the problem. We were able to theoretically confirm the efficiency observed in the experiments by showing the expected polynomial running time in the considered case. In particular, we put a lot of emphasis on justifying the choice of the distribution based on the empirical observations. The efficiency results, and in particular the average-case polynomial time result, also inherit to the very well-known elimination problem. As far as we know, this is the first average-case result in the field of sports prediction problems and one of only a handful in the field of computational social choice.

In Chapter 7, we then discussed our proposal to apply the smoothed analysis by Spielman and Teng [2004, 2009] in the area of computational social choice to both the study of the computational complexity and axiomatic properties. In particular, we reviewed the extensive related work that has appeared subsequently to our proposal, showing that smoothed analysis can be of great value for both the study of the computational complexity and the study of axiomatic properties in the field of computational social choice.

Thus, a key aspect that has driven us through this thesis has been the concept of uncertainty in relation to computational complexity: in Chapter 3, we studied the complexity of evaluating the robustness of election outcomes in the presence of uncertainty about the preferences, in Chapter 4, we studied the complexity of evaluating the robustness of election outcomes and competition outcomes in the presence of uncertainty about the scoring rule, in Chapter 5, we examined the complexity of predicting and evaluating the probabilistic robustness of election outcomes in the presence of uncertainty about the preferences, in Chapter 6, we examined the complexity of predicting sports tournaments in the presence of uncertainty about the outcomes of the remaining matches, and in Chapter 7, we discussed the impact of uncertainty on the concept of computational complexity itself.

## 8.2 Future Work & Directions

In the following, we will conclude by briefly discussing directions for future work. Since we have already discussed the specific opportunities for future work in the individual chapters, we will discuss more general directions here.

A direction that has proven extremely fruitful in the course of this thesis, but also in the literature, is the study of problems in terms of their worst-case complexity with the clear understanding that the worst-case results, and in particular potential hardness results, are a call to understand the underlying mechanisms of the problem that determine the complexity and their relevance. For this purpose, the study of the complexity in special cases, the study of the parameterized complexity, and the study of the average-case complexity have been particularly helpful.

An approach that was highly useful in this thesis was to let the research be guided by insights from experiments using real-world data and synthetic data, and to critically examine the results found using these. This approach has several advantages. For example, considering real-world data and recognizing its characteristics helps to find parameterized algorithms that provide significant practical progress. The practical and theoretical relevance of such algorithms are not mutually exclusive, e.g. our FPT algorithm for the prediction of sports tournaments in Chapter 6 showed not only that the problem can be solved much faster in practice than the worst-case hardness suggests, but also that it can be solved in polynomial time even in the worst-case in the case of at most two remaining matchdays. In addition, the implementation and application of the algorithms to real-world data, such as in Chapter 4 for the design of scoring rules, shows whether the problem is even close to being relevant in practice or whether the prerequisites are so strict that interesting use cases rarely arise in practice. While the previous comments refer to advantages of considering experiments in studies of given problems, keeping potential experiments in mind also has a conceptual advantage in planning studies and finding significant problems. Keeping meaningful experiments in mind can act as a reality check, since one cannot completely lose touch with reality and the applicability of the results in practice. Thus, we think that this approach is particularly appropriate for future studies of problems in the area of computational social choice and the study of computational problems in sports.

Another direction is the adoption of, possibly more realistic and relevant, measures of complexity than worst-case complexity. For example, the thoroughly discussed average-case complexity and smoothed analysis still offer great potential in the field of computational social choice, as we have seen in Chapter 6 and Chapter 7. But other complexity concepts, such as parallel complexity theory, which deals with the possibilities and limits of the parallelization of algorithms and problems and addresses the current demands in the context of big data, the stagnant development of the speed of single processor cores, and the focus on large computer clusters, have not yet been extensively studied in the field of computational social choice. See Csar et al. [2017] for the, to the best of our knowledge, sole results in this regard. Naturally, the rise of quantum computing, and the resulting increasing relevance of quantum complexity theory, also opens up other questions such as the protection of elections and the protection of voters' privacy.

Finally, it must be acknowledged that many of the studies discussed here are theoretical groundwork, as long as more elaborate decision making processes, such as elections with more expressive preferences than the currently used plurality voting systems, more fair allocation procedures, e.g. those developed in cake cutting for decades, and advances in tournament design, are not widely accepted and applied in practice. Thus, one of the main tasks of the field of computational social choice should be to make those future insights as well as those of the last decades accessible to the broad public, e.g., through simple explanations that build trust and accessible implementations.

# Bibliography


D. Adams. *The Hitchhiker's Guide to the Galaxy*. Pan Books, 1979.

S. Arora and B. Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009.

K. Arrow. *Social Choice and Individual Values*. John Wiley & Sons, 1951.

D. Arthur, B. Manthey, and H. Röglin. Smoothed Analysis of the $k$-Means Method. *Journal of the ACM (JACM)*, 58(5):1–31, 2011.

H. Aziz, P. Biró, R. de Haan, and B. Rastegari. Pareto Optimal Allocation Under Uncertain Preferences: Uncertainty Models, Algorithms, and Complexity. *Artificial Intelligence*, 276: 57–78, 2019.

H. Aziz, P. Biró, S. Gaspers, R. de Haan, N. Mattei, and B. Rastegari. Stable Matching with Uncertain Linear Preferences. *Algorithmica*, 82(5):1410–1433, 2020.

Y. Bachrach, N. Betzler, and P. Faliszewski. Probabilistic possible winner determination. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, pages 697–702. AAAI Press, 2010.

C. Banderier, R. Beier, and K. Mehlhorn. Smoothed Analysis of Three Combinatorial Problems. In *International Symposium on Mathematical Foundations of Computer Science*, pages 198–207. Springer, 2003.

J. Bartholdi, C. Tovey, and M. Trick. The Computational Difficulty of Manipulating an Election. *Social Choice and Welfare*, 6(3):227–241, 1989a.

J. Bartholdi, C. Tovey, and M. Trick. Voting Schemes for which It Can Be Difficult to Tell Who Won the Election. *Social Choice and Welfare*, 6(2):157–165, 1989b.

J. Bartholdi, C. Tovey, and M. Trick. How Hard Is It to Control an Election? *Mathematical and Computer Modelling*, 16(8-9):27–40, 1992.

D. Baumeister and T. Hogrebe. How Hard Is the Manipulative Design of Scoring Systems? In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 74–80, 2019a.

D. Baumeister and T. Hogrebe. Manipulative Design of Scoring Systems (Extended Abstract). In *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems*, pages 1814–1816. IFAAMAS, 2019b.

D. Baumeister and T. Hogrebe. Complexity of Election Evaluation and Probabilistic Robustness. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems*, pages 1771–1773, 2020.

D. Baumeister and T. Hogrebe. Complexity of Scheduling and Predicting Round-Robin Tournaments. In *Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems*, pages 178–186. IFAAMAS, 2021a.

D. Baumeister and T. Hogrebe. On the Complexity of Predicting Election Outcomes and Estimating Their Robustness. In *Proceedings of the 18th European Conference on Multiagent Systems*, pages 228–244. Springer, 2021b.

D. Baumeister and T. Hogrebe. On the Average-Case Complexity of Predicting Round-Robin Tournaments (Extended Abstract). In *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*. IFAAMAS, 2022.

D. Baumeister and T. Hogrebe. On the Complexity of Predicting Election Outcomes and Estimating Their Robustness. *SN Computer Science*, Submitted.

D. Baumeister and J. Rothe. Taking the Final Step to a Full Dichotomy of the Possible Winner Problem in Pure Scoring Rules. *Information Processing Letters*, 112(5):186–190, 2012.

D. Baumeister, G. Erdélyi, and J. Rothe. How Hard Is it to Bribe the Judges? A Study of the Complexity of Bribery in Judgment Aggregation. In *International Conference on Algorithmic Decision Theory*, pages 1–15. Springer, 2011a.

D. Baumeister, M. Roos, and J. Rothe. Computational Complexity of Two Variants of the Possible Winner Problem. In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems*, pages 853–860. IFAAMAS, 2011b.

D. Baumeister, M. Roos, J. Rothe, L. Schend, and L. Xia. The Possible Winner Problem with Uncertain Weights. In *Proceedings of the 20th European Conference on Artificial Intelligence*, pages 133–138, 2012.

D. Baumeister, T. Hogrebe, and L. Rey. Generalized Distance Bribery. In *Proceedings of the 7th Workshop on Computational Social Choice*, 2018.

D. Baumeister, T. Hogrebe, and L. Rey. Generalized Distance Bribery. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, pages 1764–1771. AAAI Press, 2019.

R. Beier and B. Vöcking. Random knapsack in expected polynomial time. *Journal of Computer and System Sciences*, 69(3):306–329, 2004.

S. Ben-David, B. Chor, O. Goldreich, and M. Luby. On the Theory of Average Case Complexity. *Journal of Computer and System Sciences*, 44(2):193–219, 1992.

T. Bernholt, A. Gülich, T. Hofmeister, and N. Schmitt. Football Elimination is Hard to Decide Under the 3–Point–Rule. In *International Symposium on Mathematical Foundations of Computer Science*, pages 410–418. Springer, 1999.

N. Betzler and B. Dorn. Towards a Dichotomy of Finding Possible Winners in Elections Based on Scoring Rules. *Journal of Computer and System Sciences*, 76(8):812–836, 2010.

N. Betzler, R. Bredereck, and R. Niedermeier. Theoretical and Empirical Evaluation of Data Reduction for Exact Kemeny Rank Aggregation. *Autonomous Agents and Multi-Agent Systems*, 28(5):721–748, 2014.

M. Bläser and B. Manthey. Smoothed Complexity Theory. In *Proceedings of the 37th International Symposium on Mathematical Foundations of Computer Science*, pages 198–209. Springer, 2012.

N. Boehmer, R. Bredereck, P. Faliszewski, and R. Niedermeier. Winner robustness via swap- and shift-bribery: Parameterized counting complexity and experiments. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence*, pages 52–58, 2021a.

N. Boehmer, R. Bredereck, K. Heeger, and R. Niedermeier. Bribery and Control in Stable Marriage. *Journal of Artificial Intelligence Research*, 71:993–1048, 2021b.

A. Bogdanov and L. Trevisan. Average-Case Complexity. *arXiv preprint cs/0606037*, 2006.

C. Boutilier, I. Caragiannis, S. Haber, T. Lu, A. Procaccia, and O. Sheffet. Optimal social choice functions: A utilitarian view. *Artificial Intelligence*, 227:190–213, 2015.

F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. Procaccia. *Handbook of Computational Social Choice*. Cambridge University Press, 2016.

R. Bredereck, J. Chen, P. Faliszewski, A. Nichterlein, and R. Niedermeier. Prices matter for the parameterized complexity of shift bribery. *Information and Computation*, 251:140–164, 2016.

R. Bredereck, P. Faliszewski, A. Kaczmarczyk, R. Niedermeier, P. Skowron, and N. Talmon. Robustness Among Multiwinner Voting Rules. In *International Symposium on Algorithmic Game Theory*, pages 80–92. Springer, 2017.

R. Bredereck, P. Faliszewski, A. Kaczmarczyk, R. Niedermeier, P. Skowron, and N. Talmon. Robustness Among Multiwinner Voting Rules. *Artificial Intelligence*, 290, 2021.

E. Brelsford, P. Faliszewski, E. Hemaspaandra, H. Schnoor, and I. Schnoor. Approximability of Manipulating Elections. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence*, pages 44–49, 2008.

G. Brightwell and P. Winkler. Counting linear extensions. *Order*, 8(3):225–242, 1991.

D. Burka, C. Puppe, L. Szepesváry, and A. Tasnádi. And the winner is... Chevalier de Borda: Neural networks vote according to Borda's Rule. In *Proceedings of the 6th International Workshop on Computational Social Choice*, 2016.

D. Burka, C. Puppe, L. Szepesváry, and A. Tasnádi. Voting: a machine learning approach. *European Journal of Operational Research*, 2021.

I. Caragiannis and K. Fehrs. The Complexity of Learning Approval-Based Multiwinner Voting Rules. *arXiv preprint arXiv:2110.00254*, 2021.

I. Caragiannis, X. Chatzigeorgiou, G. Krimpas, and A. Voudouris. Optimizing positional scoring rules for rank aggregation. *Artificial Intelligence*, 267:58–77, 2019.

D. Cary. Estimating the Margin of Victory for Instant-Runoff Voting. In *Proceedings of the Electronic Voting Technology Workshop and the Workshop on Trustworthy Elections*, 2011.

K. Cechlárová, E. Potpinková, and I. Schlotter. Refining the complexity of the sports elimination problem. *Discrete Applied Mathematics*, 199:172–186, 2016.

V. Chakraborty and P. Kolaitis. Classifying the Complexity of the Possible Winner Problem on Partial Chains. In *Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems*, pages 297–305, 2021.

Y. Chevaleyre, J. Lang, N. Maudet, and J. Monnot. Possible Winners when New Candidates Are Added: The Case of Scoring Rules. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, pages 762–767, 2010.

R. Christian, M. Fellows, F. Rosamond, and A. Slinko. On Complexity of Lobbying in Multiple Referenda. *Review of Economic Design*, 11(3):217–224, 2007.

A. Church. An Unsolvable Problem of Elementary Number Theory. *American Journal of Mathematics*, 58(2):345–363, 1936.

C. Colbourn. Embedding Partial Steiner Triple Systems Is NP-Complete. *Journal of Combinatorial Theory, Series A*, 35(1):100–105, 1983.

V. Conitzer and T. Walsh. Barriers to Manipulation in Voting. In F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. Procaccia, editors, *Handbook of Computational Social Choice*, chapter 6, pages 127–145. Cambridge University Press, 2016.

V. Conitzer, T. Sandholm, and J. Lang. When Are Elections with Few Candidates Hard to Manipulate? *Journal of the ACM (JACM)*, 54(3):14, 2007.

V. Conitzer, M. Rognlie, and L. Xia. Preference Functions That Score Rankings and Maximum Likelihood Estimation. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pages 109—-115, 2009.

S. Cook. The Complexity of Theorem-Proving Procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, pages 151—158. Association for Computing Machinery, 1971.

T. Csar, M. Lackner, R. Pichler, and E. Sallinger. Winner Determination in Huge Elections with MapReduce. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, 2017.

G. Cybenko. Approximation by Superpositions of a Sigmoidal Function. *Mathematics of Control, Signals, and Systems*, 2(4):303–314, 1989.

G. D'Angelo, E. Delfaraz, and H. Gilbert. Computation and Bribery of Voting Power in Delegative Simple Games. *arXiv preprint arXiv:2104.03692*, 2021.

P. Dey. Local Distance Restricted Bribery in Voting. In *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems*, pages 1925–1927, 2019.

P. Dey, N. Misra, and Y. Narahari. Frugal Bribery in Voting. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*, pages 2466–2472, 2016.

C. Dodgson. A Method for Taking Votes on More than Two Issues. 1876.

B. Dorn and I. Schlotter. Multivariate Complexity Analysis of Swap Bribery. *Algorithmica*, 64(1):126–151, 2012.

R. Downey and M. Fellows. Fixed-Parameter Tractability and Completeness. *Congressus Numerantium*, pages 161–161, 1992.

R. Downey and M. Fellows. *Parameterized Complexity.* Springer Science & Business Media, 2012.

J. Duggan and T. Schwartz. Strategic manipulability without resoluteness or shared beliefs: Gibbard-Satterthwaite generalized. *Social Choice and Welfare*, 17(1):85–93, 2000.

E. Elkind and G. Erdélyi. Manipulation Under Voting Rule Uncertainty. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, pages 627–634. IFAAMAS, 2012.

E. Elkind and P. Faliszewski. Approximation Algorithms for Campaign Management. In *International Workshop on Internet and Network Economics*, pages 473–482. Springer, 2010.

E. Elkind, P. Faliszewski, and A. Slinko. Swap Bribery. In *Proceedings of the 2nd International Symposium on Algorithmic Game Theory*, pages 299–310. Springer-Verlag *Lecture Notes in Computer Science #5814*, 2009.

E. Elkind, P. Faliszewski, S. Gupta, and S. Roy. Algorithms for Swap and Shift Bribery in Structured Elections. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems*, pages 366–374, 2020.

U. Endriss. *Trends in Computational Social Choice.* lulu.com, 2017.

M. Englert, H. Röglin, and B. Vöcking. Worst Case and Probabilistic Analysis of the 2-Opt Algorithm for the TSP. *Algorithmica*, 68(1):190–264, 2014.

G. Erdélyi, C. Reger, and Y. Yang. The Complexity of Bribery and Control in Group Identification. In *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems*, pages 1142–1150, 2017.

G. Erdélyi, C. Reger, and Y. Yang. The Complexity of Bribery and Control in Group Identification. *Autonomous Agents and Multi-Agent Systems*, 34(1):1–31, 2020.

M. Etscheid and H. Röglin. Smoothed Analysis of Local Search for the Maximum-Cut Problem. *ACM Transactions on Algorithms (TALG)*, 13(2):1–12, 2017.

P. Faliszewski. Nonuniform bribery. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 1569–1572. IFAAMAS, 2008.

P. Faliszewski and J. Rothe. Control and Bribery in Voting. In F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. Procaccia, editors, *Handbook of Computational Social Choice*, chapter 7, pages 146–168. Cambridge University Press, 2016.

P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. The Complexity of Bribery in Elections. In *Proceedings of the 21st AAAI Conference on Artificial Intelligence*, pages 641–646, 2006.

P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. How Hard Is Bribery in Elections? *Journal of Artificial Intelligence Research*, 35:485–532, 2009a.

P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Llull and Copeland Voting Computationally Resist Bribery and Constructive Control. *Journal of Artificial Intelligence Research*, 35:275–341, 2009b.

P. Faliszewski, P. Skowron, and N. Talmon. Bribery as a Measure of Candidate Success: Complexity Results for Approval-Based Multiwinner Rules. In *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems*, pages 6–14, 2017.

P. Faliszewski, S. Szufa, and N. Talmon. Optimization-Based Voting Rule Design: The Closer to Utopia the Better. In *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems*, pages 32–40. IFAAMAS, 2018.

P. Faliszewski, P. Manurangsi, and K. Sornat. Approximation and Hardness of Shift-Bribery. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, pages 1901–1908, 2019.

P. Faliszewski, P. Manurangsi, and K. Sornat. Approximation and Hardness of Shift-Bribery. *Artificial Intelligence*, 298, 2021.

P. Faliszewski, S. Szufa, and N. Talmon. Optimization-Based Voting Rule Design: The Closer to Utopia the Better. In *Collective Decisions: Theory, Algorithms And Decision Support Systems*, pages 17–51. Springer, 2022.

B. Fazzinga, S. Flesca, and F. Parisi. On the Complexity of Probabilistic Abstract Argumentation Frameworks. *ACM Transactions on Computational Logic (TOCL)*, 16(3):1–39, 2015.

Z. Fitzsimmons and E. Hemaspaandra. The Complexity of Succinct Elections. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, 2017.

B. Flanigan, D. Halpern, and A. Psomas. A Framework for Smoothed Analysis of Social Choice. 2022. URL https://daniel-halpern.com/files/smoothed-voting.pdf.

L. Ford and D. Fulkerson. Maximal Flow Through a Network. *Canadian Journal of Mathematics*, 8:399–404, 1956.

R. Freeman, M. Brill, and V. Conitzer. General Tiebreaking Schemes for Computational Social Choice. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems*, pages 1401–1409, 2015.

M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.

A. Gibbard. Manipulation of Voting Schemes: A General Result. *Econometrica: Journal of the Econometric Society*, pages 587–601, 1973.

K. Gödel. On Undecidable Propositions of Formal Mathematical Systems. Lecture Notes by S. Kleene and J. Rosser. *Institute for Advanced Study, Princeton, NJ*, pages 39–74, 1934.

U. Grandi and P. Turrini. A Network-Based Rating System and Its Resistance to Bribery. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*, pages 301–307, 2016.

D. Gusfield and C. Martel. A Fast Algorithm for the Generalized Parametric Minimum Cut Problem and Applications. *Algorithmica*, 7(1):499–519, 1992.

J. Hartmanis and R. Stearns. On the Computational Complexity of Algorithms. *Transactions of the American Mathematical Society*, 117:285–306, 1965.

N. Hazon, Y. Aumann, S. Kraus, and M. Wooldridge. On the evaluation of election outcomes under uncertainty. *Artificial Intelligence*, 189:1–18, 2012.

E. Hemaspaandra and H. Schnoor. Dichotomy for Pure Scoring Rules Under Manipulative Electoral Actions. In *Proceedings of the 22nd European Conference on Artificial Intelligence*, pages 1071–1079, 2016.

E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Exact Analysis of Dodgson Elections: Lewis Carroll's 1876 Voting System is Complete for Parallel Access to NP. *Journal of the ACM*, 44(6):806–825, 1997.

E. Hemaspaandra, H. Spakowski, and J. Vogel. The Complexity of Kemeny Elections. *Theoretical Computer Science*, 349(3):382–391, 2005.

L. Hemaspaandra. Computational Social Choice and Computational Complexity: BFFs? In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pages 7971–7977. AAAI Press, 2018.

D. Hilbert and W. Ackermann. *Grundzüge der Theoretischen Logik*. J. Springer, 1928.

A. Hoffman and T. Rivlin. When is a team "mathematically" eliminated? In *Proceedings of the Princeton Symposium on Mathematical Programming*, pages 391—401. Princeton University Press, 1967.

I. Holyer. The NP-Completeness of Edge-Coloring. *SIAM Journal on Computing*, 10(4): 718–720, 1981.

R. Impagliazzo. A Personal View of Average-Case Complexity. In *Proceedings of Structure in Complexity Theory. Tenth Annual IEEE Conference*, pages 134–147. IEEE, 1995.

M. Kalech, S. Kraus, G. Kaminka, and C. Goldman. Practical voting rules with partial information. *Autonomous Agents and Multi-Agent Systems*, 22(1):151–182, 2011.

R. Karp. Reducibility Among Combinatorial Problems. In *Complexity of Computer Computations*, pages 85–103. Springer, 1972.

J. Kemeny. Mathematics Without Numbers. *Daedalus*, 88(4):577–591, 1959.

W. Kern and D. Paulusma. The computational complexity of the elimination problem in generalized sports competitions. *Discrete Optimization*, 1(2):205–214, 2004.

S. Kleene. General recursive functions of natural numbers. *Mathematische Annalen*, 112(1): 727–742, 1936.

D. Knop, M. Koutecký, and M. Mnich. Voting and Bribing in Single-Exponential Time. *ACM Transactions on Economics and Computation (TEAC)*, 8(3):1–28, 2020.

K. Konczak and J. Lang. Voting procedures with incomplete preferences. In *Proceedings of the IJCAI-05 Multidisciplinary Workshop on Advances in Preference Handling*, volume 20. Citeseer, 2005.

J. Kruger and S. Schneckenburger. Fall if it lifts your teammate: a novel type of candidate manipulation. In *Proceedings of the 18th International Conference on Autonomous Agents and Multiagent Systems*, pages 1431–1439, 2019.

R. Kumar and S. Vassilvitskii. Generalized distances between rankings. In *Proceedings of the 19th International Conference on World Wide Web*, pages 571–580. ACM Press, 2010.

J. Lang. Collective Decision Making under Incomplete Knowledge: Possible and Necessary Solutions. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence*, pages 4885–4891, 2020.

F. Lehoucq. Electoral fraud: Causes, Types, and Consequences. *Annual Review of Political Science*, 6(1):233–256, 2003.

L. Levin. Average Case Complete Problems. *SIAM Journal on Computing*, 15(1):285–286, 1986.

A. Lin. *Solving Hard Problems in Election Systems.* Rochester Institute of Technology, 2012.

A. Lintott. Electoral Bribery in the Roman Republic. *The Journal of Roman Studies*, 80: 1–16, 1990.

L. Lovasz. Coverings and Colorings of Hypergraphs. In *Proceedings of the 4th Southeastern Conference on Combinatorics, Graph Theory, and Computing*, pages 3–12, 1973.

T. Magrino, R. Rivest, E. Shen, and D. Wagner. Computing the Margin of Victory in IRV Elections. In *Proceedings of the Electronic Voting Technology Workshop/Workshop on Trustworthy.* USENIX Association, 2011.

C. Mallows. Non-null ranking models. *Biometrika*, 44:114–130, 1957.

B. Manthey and H. Röglin. Smoothed Analysis: Analysis of Algorithms Beyond Worst Case. *IT-Information Technology*, 53(6):280–286, 2011.

N. Mattei and T. Walsh. PrefLib: A Library for Preferences http://www.preflib.org. In *Proceedings of the 3rd International Conference on Algorithmic Decision Theory*, pages 259–270. Springer-Verlag *Lecture Notes in Artificial Intelligence #8176*, 2013.

N. Mattei, J. Goldsmith, A. Klapper, and M. Mundhenk. On the complexity of bribery and manipulation in tournaments with uncertain information. *Journal of Applied Logic*, 13(4): 557–581, 2015.

C. Maushagen, M. Neveling, J. Rothe, and A. Selker. Complexity of Shift Bribery in Iterative Elections. In *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems*, page 1567–1575, 2018.

S. McCormick. Fast Algorithms for Parametric Scheduling Come from Extensions to Parametric Maximum Flow. *Operations Research*, 47(5):744–756, 1999.

A. Meyer and L. Stockmeyer. The Equivalence Problem for Regular Expressions With Squaring Requires Exponential Space. In *Proceedings of the 13th IEEE Symposium on Switching and Automata Theory*, pages 125–129, 1972.

M. Mihail and P. Winkler. On the Number of Eulerian Orientations of a Graph. *Algorithmica*, 16(4):402–414, 1996.

J. Myhill. Linear Bounded Automata. *Wright Air Development Division*, 1960.

G. Nemhauser and Z. Ullmann. Discrete Dynamic Programming and Capital Allocation. *Management Science*, 15(9):494–505, 1969.

S. Neumann and A. Wiese. This House Proves That Debating is Harder Than Soccer. In *Proceedings of the 8th International Conference on Fun with Algorithms*, 2016.

M. Neveling, J. Rothe, and R. Weishaupt. The Possible Winner Problem with Uncertain Weights Revisited. In *International Symposium on Fundamentals of Computation Theory*, pages 399–412. Springer, 2021.

S. Obraztsova and E. Elkind. Optimal Manipulation of Voting Rules. In *Proceedings of the 11th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 619–626. IFAAMAS, 2012.

D. Pálvölgyi. Deciding Soccer Scores and Partial Orientations of Graphs. *Tech. Rep. TR-2008-03, Egerváry Research Group, Institute of Mathematics, Eötvös Loránd University, Budapest, Hungary*, 2009.

C. Papadimitriou. *Computational Complexity.* Addison-Wesley, 1994.

A. Procaccia, A. Zohar, Y. Peleg, and J. Rosenschein. The learnability of voting rules. *Artificial Intelligence*, 173(12-13):1133–1149, 2009.

R. Rasmussen and M. Trick. Round robin scheduling – a survey. *European Journal of Operational Research*, 188(3):617–636, 2008.

A. Rey and J. Rothe. Bribery in Path-Disruption Games. In *International Conference on Algorithmic Decision Theory*, pages 247–261. Springer, 2011.

A. Rosa and W. Wallis. Premature Sets of 1-Factors or How Not to Schedule Round Robin Tournaments. *Discrete Applied Mathematics*, 4(4):291–297, 1982.

J. Rothe, editor. *Economics and Computation, An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division.* Springer, 2016.

J. Rothe, D. Baumeister, and T. Hogrebe. Towards Reality: Smoothed Analysis in Computational Social Choice. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems*, pages 1691–1695, 2020.

D. Ryvkin and A. Ortmann. The Predictive Power of Three Prominent Tournament Formats. *Management Science*, 54(3):492–504, 2008.

S. Saarinen, J. Goldsmith, and C. Tovey. Probabilistic Copeland Tournaments. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pages 1851–1852, 2015.

M. 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.

B. Schwartz. Possible Winners in Partially Completed Tournaments. *SIAM Review*, 8(3): 302–308, 1966.

D. Shiryaev, L. Yu, and E. Elkind. On elections with robust winners. In *Proceedings of the 13th International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 415–422. IFAAMAS, 2013.

D. Spielman and S. Teng. Smoothed Analysis of Algorithms: Why the Simplex Algorithm Usually Takes Polynomial Time. *Journal of the ACM*, 51(3):385–463, 2004.

D. Spielman and S. Teng. Smoothed Analysis: An Attempt to Explain the Behavior of Algorithms in Practice. *Communications of the ACM*, 52(10):76–84, 2009.

S. Toda. PP is as hard as the polynomial-time hierarchy. *SIAM Journal on Computing*, 20 (5):865–877, 1991.

D. Tokaji. The Paperless Chase: Electronic Voting and Democratic. *Fordham L. Rev.*, 73: 1711, 2004.

A. Turing. Computability and $\lambda$-Definability. *The Journal of Symbolic Logic*, 2(4):153–163, 1937a.

A. Turing. On Computable Numbers, with an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, s2-42(1):230–265, 1937b.

I. Urken. *Classics of Social Choice*. University of Michigan Press, 1995.

L. Valiant. The Complexity of Computing the Permanent. *Theoretical computer science*, 8 (2):189–201, 1979.

L. Valiant. A Theory of the Learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

A. Vattani. $k$-means Requires Exponentially Many Iterations Even in the Plane. *Discrete & Computational Geometry*, 45(4):596–616, 2011.

P. Viappiani. Positional Scoring Rules with Uncertain Weights. In *International Conference on Scalable Uncertainty Management*, pages 306–320. Springer, 2018.

V. Vizing. On an estimate of the chromatic class of a $p$-graph. *Discret Analiz*, 3:25–30, 1964.

K. Wayne. A New Property and a Faster Algorithm for Baseball Elimination. *SIAM Journal on Discrete Mathematics*, 14(2):223–229, 2001.

V. Williams. Knockout Tournaments. In F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. Procaccia, editors, *Handbook of Computational Social Choice*, chapter 19, pages 453–474. Cambridge University Press, 2016.

L. Xia. Computing the margin of victory for various voting rules. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pages 982–999. ACM Press, 2012.

L. Xia. The Smoothed Possibility of Social Choice. *Advances in Neural Information Processing Systems*, 33, 2020.

L. Xia. How Likely Are Large Elections Tied? In *Proceedings of the 22nd ACM Conference on Economics and Computation*, pages 884–885, 2021.

L. Xia and W. Zheng. The Smoothed Complexity of Computing Kemeny and Slater Rankings. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, pages 5742–5750, 2021.

R. Yager. On Ordered Weighted Averaging Aggregation Operators in Multicriteria Decision-making. *IEEE Transactions on systems, Man, and Cybernetics*, 18(1):183–190, 1988.

Y. Yang, Y. Shrestha, and J. Guo. How Hard Is Bribery with Distance Restrictions. In *Proceedings of the 22nd European Conference on Artificial Intelligence*, pages 363–371. IOS Press, 2016.

A. Zagoury, O. Keller, A. Hassidim, and N. Hazon. Targeted Negative Campaigning: Complexity and Approximations. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence*, pages 5768–5778, 2021.

A. Zhou and J. Guo. Parameterized Complexity of Shift Bribery in Iterative Elections. In *Proceedings of the 19th International Conference on Autonomous Agents and Multiagent Systems*, pages 1665–1673, 2020.

W. Zwicker. Introduction to the Theory of Voting. In F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. Procaccia, editors, *Handbook of Computational Social Choice*, chapter 2, pages 23–56. Cambridge University Press, 2016.

# Eidesstattliche Erklärung

entsprechend §5 der Promotionsordnung vom 15.06.2018.

Ich versichere an Eides Statt, dass die Dissertation von mir selbständig und ohne unzulässige fremde Hilfe unter Beachtung der „Grundsätze zur Sicherung guter wissenschaftlicher Praxis an der Heinrich-Heine-Universität Düsseldorf“ erstellt worden ist. Desweiteren erkläre ich, dass ich eine Dissertation in der vorliegenden oder in ähnlicher Form noch bei keiner anderen Institution eingereicht habe.

Teile dieser Dissertation wurden bereits in Form folgender Zeitschriftenartikel und Konferenzberichte veröffentlicht oder zur Begutachtung eingereicht und sind entsprechend gekennzeichnet: Baumeister et al. [2018], Baumeister et al. [2019], Baumeister and Hogrebe [2019a], Baumeister and Hogrebe [2019b], Baumeister and Hogrebe [2020], Baumeister and Hogrebe [2021a], Baumeister and Hogrebe [2021b], Baumeister and Hogrebe [2022], and Baumeister and Hogrebe [Submitted].

| | |
|---|---|
| Ort, Datum | Tobias Alexander Hogrebe |