
# Deciding the Uncertain: Axiomatic Aspects of Fair Division and Computational Complexity Studies in Computational Social Choice and Graph Theory

Inaugural-Dissertation

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

vorgelegt von

**Robin Weishaupt**
aus Duisburg

Düsseldorf, März 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. Herr Prof. Dr. Jörg Rothe
2. Frau apl. Prof. Dr. Dorothea Baumeister

Tag der mündlichen Prüfung: 30. Mai 2022

# Abstract

In this work we study uncertainty across four different areas of theoretical computer science. For three of the four areas, we analyze the hardness of new and existing problems with tools from computational complexity theory. For the fourth area, we inspect its fundamental axioms, identify discrepancies, and suggest improvements. The four areas we address are judgment aggregation, preference aggregation by voting, fair division of divisible goods, and stability of graphs.

Judgment aggregation studies the aggregation of individual judges' judgments over a given agenda via some aggregation rule into a collective outcome. In our work we study a generalized family of sequential judgment aggregation rules. A sequential judgment aggregation rule aggregates a set of judgments into a collective outcome by following a specified order over the elements of the agenda. With this family of rules defined, we determine the computational complexity of several uncertainty-related decision problems for these rules. Among the studied problems are the classical winner problem as well as problems addressing manipulability.

In preference aggregation by voting, we are given a set of candidates and a set of voters expressing their preference orders over the candidates. Using a predefined voting rule, we determine one or multiple winners for such an election by aggregating the voters' preferences. In our work, we introduce a novel variant of the possible winner problem, the possible winner with uncertain weights problem. In this problem one is given a set of candidates, a distinguished candidate, and a set of votes over the candidates with uncertain weights. One is asked whether there is a weight allocation to the votes such that the distinguished candidate wins the weighted election. We define a framework to study the computational complexity of the problem and three further variants for integer and rational weights as well as several voting rules.

The area of fair division of divisible goods is also called cake-cutting. Its general setting consists of an arbitrarily divisible good and a set of players with preferences over the good, willing to split the good among themselves. Giving the area its name, a cake is used as a metaphor for the good and an interactive cake-cutting procedure portions the cake among the players. In our work, we study basic axioms of cake-cutting: What pieces of cake should be considered as valid pieces, such that every player can evaluate them. We survey current cake-cutting literature, identify discrepancies, and make suggestions supported by the means of measure theory.

Stability of graphs belongs to the area of graph theory. Given a graph and a graph parameter, a graph is called stable with respect to this parameter if removing any single edge of the graph does not alter the parameter's value. Similar concepts are defined for vertices and adding edges and vertices. We build on already defined decision problems in this field and study the complexity of these stability related problems for special graph classes. In total, we cover the graph classes of trees, forests, bipartite graphs, and co-graphs as well as four well-known graph parameters.

# Acknowledgments

After successfully completing my studies in business administration at WHU – Otto Beisheim School of Management, I came to Heinrich-Heine-Universität Düsseldorf to follow my passion for mathematics and computer science by studying both subjects and starting from scratch as an undergraduate student. Finishing these undergraduate studies, I was given the exceptional opportunity to immediately begin my doctorate studies via the university's fast-track program. It would be presumptuous to pretend that all these achievements are solely my own and hence, at this point I want to express my deepest gratitude to

**Jörg** for accepting me as a fast-track doctoral student, having and giving me the confidence in my abilities, and for providing his restless support day and night throughout all corners of the academic world. Without his endorsement, this thesis and my doctorate studies would not have been possible.

**Dorothea** for being my mentor, guiding me through phases of doubt and uncertainty, and her willingness to help in any situation.

**my colleagues** for becoming my co-authors, the intense discussions on- and off-topic, and the great collegial atmosphere that made this time so enjoyable.

**Linus** for proofreading this dissertation, becoming my co-author, and all the constructive work we did together.

**Simon** as one of the great friends I made along the way at Heinrich-Heine-Universität Düsseldorf for sharing my passion for Computer Science and everything that came out of our shared passion.

**Marc and Niklas** as my closest friends outside the university for always being around and for the countless coffees, beers, and calls we had, discussing what had to be discussed.

**my family** for paving my way, enabling me to follow my interests. Without their everlasting love, care, and guidance I would not be who I am today.

Robin

# Table of Contents

# 1 Introduction

In this work, we try to decide the uncertain. This statement left alone is, of course, much too bold and beyond our scope, such that it needs to be put into some perspective: Computer science is an extremely broad and diversified field with a vast amount of different subareas, all addressed by current research. Each of these subareas contains open problems with unanswered questions that wait to be settled. Approaching the initial statement from this perspective, in this work we concentrate on the wider subarea of theoretical computer science and present several new results, proving unknown facts, and so helping to decide the uncertain. However, this explanation alone probably does not vindicate the title of our work. Instead, despite proving new results and pushing the frontier of knowledge in the area of theoretical computer science an imperceptible inch forward, all of the results that we address in the ensuing chapters are somehow internally related to uncertainty. In our work we address four different subareas of computer science, namely judgment aggregation, preference aggregation by voting, fair division of divisible goods, also known as cake-cutting, and graph theory. In every of these four subareas, except for cake-cutting, we analyze problems with the tools of computational complexity theory in order to determine how hard it is to compute solutions to these problems. For cake-cutting, we follow a different approach and instead study the very foundations of the field, explicitly formulating new suggestions for some of the basic axioms in the area.

Before outlining the four different subareas that we cover in our work, we want to describe the aforementioned tools from computational complexity theory which we make use of. Computational complexity theory itself can be considered as one of the very fundamental subareas in theoretical computer science research [119]. This field is mainly occupied with determining for given problems how hard they are to solve [126]. To do so, standardized, formal methods and tools to study the hardness of problems have been introduced. Giving the field its name, the hardness of a problem is also known as the problem's computational complexity. A problem's computational complexity can be measured along different dimensions such as time and space. In other words, the means of computational complexity enable us to formulate statements about what amount of time or space is required to solve a given problem. Furthermore, this unit of complexity is unified in such a way that we can compare the complexities of different problems, allowing us to say whether a given problem is more, less, or equally as complex as some other problem. Based on the time or space required to solve a given problem, we can categorize problems into different complexity classes, each containing problems of similar complexity along a given dimension [134].

Previously, we mentioned the areas of judgment aggregation and preference aggregation by voting. Both areas belong to the field of computational social choice (COMSOC). COMSOC is, compared to other areas such as complexity theory, a relatively young subject of theoretical computer science research and has its origins in the field of social choice theory [32]. While social choice theory addresses problems like

aggregating multiple individual preferences or judgments into collective outcomes, COMSOC approaches these questions from a computer science perspective [3, 127]. In doing so, COMSOC extends the typical social choice theoretic questions by a computational aspect and, for example, studies runtimes of preference aggregation mechanisms, their manipulability by untruthfully voting agents or external bribers, as well as the invention of new aggregation procedures satisfying certain complexity and axiomatic requirements. Over the past years, COMSOC has become a central sub field of artificial intelligence and multiagent-systems [88]. Research results have diverse applications in economical contexts, politics, technologies, and basically all situations where collective decision-making is required.

The first subarea of COMSOC that we study in our work is the area of judgment aggregation. Judgment aggregation addresses scenarios where several judges with individual judgments over a set of possibly logically interconnected boolean formulas want to aggregate these judgments into a collective outcome, representing the overall preferences of the group as well as possible [53]. We define a family of judgment aggregation rules that can be used by judges to aggregate their individual judgments into a collective outcome. Having formulated this family of rules, we study the rules' properties and confirm that the rules are consistent, i.e., they never return an inconsistent or contradictory outcome as aggregated judgment. Furthermore, we determine the complexity of executing such a judgment aggregation rule for a given set of judgments and show that the defined family of rules is rather resistant against manipulation attempts. Finally, we relate the family of judgment aggregation rules to other, already existing rules from the literature, enabling the transfer of our established complexities to further rules.

The second subarea of COMSOC which we study is the area of preference aggregation by voting. This area addresses elections similar to ones one might know from politics. The general scenario consists of a set of candidates as well as a set of voters providing preference orders, also called votes, over the candidates. Given a voting rule, one then determines one or multiple winners of the election [22]. A very famous problem in the area of preference aggregation by voting is the possible winner problem [98]. In this problem one is given a set of candidates, a distinguished candidate, as well as a set of partial preferences over the candidates, i.e., preference orders in which not all candidates occur, and is asked whether there is an extension of the partial votes to total ones such that the distinguished candidate is a winner of the election. In our work, we introduce a novel variant of the possible winner problem, namely the possible winner with uncertain weights problem. This variant addresses weighted elections, i.e., elections where the voters' preferences have weights. Thereby, all votes are total preferences over the candidates but some of the preferences' weights are unknown. The question posed is whether there exists a weight allocation to the votes without weights such that the distinguished candidate is a winner of the election. Having introduced this problem, we introduce three further variants of the problem as well as a framework to study all variants for nonnegative integer and rational weights. With the framework at hand, we resolve all variants' computational complexities for several well-known voting rules.

Having covered these two areas of COMSOC, we turn to a field of theoretical computer science closely related to COMSOC, namely fair division of divisible goods, also called cake-cutting [109]. The basic setting of this area consists of a set of players that want to share a heterogeneous, arbitrarily divisible good among each other. Usually, this good is metaphorically illustrated by a cake, as a cake is arbitrarily divisible and can be heterogeneous, e.g., when some parts possess vanilla flavors and others do not. Every player has some secret valuation function for the cake and all players follow a cake-cutting protocol to share the cake among each other. In this work we address the very foundations of cake-cutting and its axioms. In particular, we discuss what pieces of cake should be allowed to be cut during the execution of a cake-cutting protocol and the implications of our suggestions on valuation functions and the practical feasibility of cake-cutting protocols.

Finally, our work turns to the area of graph theory and studies the stability of graphs. Graph theory itself is one of the oldest areas of research in computer science and dates back to the work by Euler [61] in 1741. Given a graph and a graph parameter, we call the graph stable with respect to this parameter if the value of the parameter for the graph does not change by removing any single edge of the graph. Besides this concept of edge-stability, there are also the concepts of vertex-stability, edge-unfrozenness, and vertex-unfrozenness, where the last two notions relate to adding to, instead of deleting from, a graph. Building on previous results by Frei, Hemaspaandra, and Rothe [67], we study the stability of graphs for special graph classes. Doing so, we introduce algorithms for these classes that enable to efficiently determine whether a given graph belonging to one of the studied classes is stable or not. In our work we cover four of the most known graph parameters, the independence set number, the clique number, the vertex cover number, and the chromatic number as well as four graph classes, trees, forests, bipartite graphs, and co-graphs.

Overall, this work is structured into seven chapters including this introduction. The second chapter provides exhaustive background information on all research areas of computer science that we address in our work. We provide introductions to computational complexity, graph theory, judgment aggregation, preference aggregation by voting, as well as cake-cutting. For each of these areas we explain all technical notions and notations required throughout our work, illustrate these with small examples, and present recent as well as historically relevant literature. In Chapter 3, we present our results related to judgment aggregation. Chapter 4 addresses our work on preference aggregation by voting, Chapter 5 introduces our work on cake-cutting, and Chapter 6 describes our results regarding the stability of graphs. We end this work with a conclusion, giving a brief summary as well as an outlook on possible future work.

# 2 Background

In this chapter of our work we introduce all areas of research that are either used as preliminaries or are affected by our results as established in later chapters. Besides providing overviews of these fields, we present central ideas, notions, and concepts required to be able to understand all results and to follow corresponding proofs. In the first section we introduce the research area of computational complexity theory. As the title of our work already indicates, this area forms a fundamental basis for our research and its tools are used across almost all our results. The second section addresses the wide field of graph theory. Besides a thorough introduction to the concepts of graph theory, as well as some advanced concepts therein, we build on the previous section and present a select number of notions by combining computational complexity theory with graph theory. The third section thematizes the research areas of computational social choice and fair division. Research in computational social choice is strongly diversified and contains a large range of different subfields. Besides a short, general overview of the area of computational social choice, we provide two parts covering the fields of judgment aggregation and of preference aggregation by voting. The third and last part addresses the area of cake-cutting, a subfield of fair division.

## 2.1 Computational Complexity Theory

In this section we provide an overview of the field of computational complexity theory. For a more in-depth introduction, we recommend the books by Arora and Barak [2], Papadimitriou [119], and Rothe [126], all providing an exhaustive overview of the field of computational complexity theory. As implied by our introduction in Chapter 1, our work mainly focuses on determining how hard it is to find solutions to certain problems and to classify the problems accordingly. Thereby, the colloquial formulation "how hard" needs to be specified in order to obtain a standardized, comparable measure for the complexity of problems. To do so, we make use of the research area of computational complexity theory. This area of research, whose origin is usually linked to the work by Hartmanis and Stearns [84] in 1965, provides mathematically formulated, precise means to classify the computational complexity of problems. The most fundamental building block within this area is the *Turing machine*, introduced by Turing [136] in 1936, used as an abstract model to simulate computations in a formalized way. In order to make the computational complexity of problems comparable, one must agree on a dimension along which one wants to compare complexity. The two most commonly used dimensions are *time*, i.e., the number of computation steps required, and *space*, i.e., the amount of storage required during a computation. In this work, we restrict ourselves to comparing problems solely with respect to computation time as a measure of complexity.

Based on this computation model of a Turing machine and one of the two dimensions to compare along, we are almost set to determine the computational complexity of problems. However, we also need to specify how problems, whose computational complexity shall be determined, should be defined. In the research area of computa-

tional complexity there are several different types of problems that can be analyzed, e.g., *decision problems*, *optimization problems*, *search problems*, and within our work, we study decision problems. A decision problem is defined by two parts, a specification for the input, i.e., how a concrete problem instance looks like, and a yes-no question that shall be decided for the given problem instance. Subsequently, we provide an example of a well-known and well-studied decision problem. Garey and Johnson [71] count this problem as one of the six problems belonging to the "basic core" of decision problems used in computational complexity theory.[1]

| Vertex Cover (VC) | |
|---|---|
| **Given:** | A graph $G = (V, E)$ and an integer $k \in \mathbb{N}$. |
| **Question:** | Is there a vertex cover of size $k$ or less for $G$? |

If for a given instance the answer to the problem is yes, we call the instance a YES-*instance*. Otherwise, if the question is answered with no, we call the instance a NO-*instance*. To measure the complexity of such a decision problem, we need to formulate a program that can be executed by a Turing machine and is able to solve every instance of this problem. Formulating such programs is a technically involved task. Fortunately, assuming the Church-Turing thesis [69], which states that every practically executable algorithm can be computed by a Turing machine, it is sufficient to describe an algorithm solving the problem in a more high-level and abstract language. Consequently, the task transforms from formulating a Turing machine to formulating an algorithm that is able to answer every instance of the given decision problem.[2] Then, the question is how many computation steps are required to solve a given instance of this problem. Of course, an absolute answer, for example ten computation steps, does not make sense, as usually the number of computation steps an algorithm requires to solve an input instance depends on the size of the instance. When we address the *size* of an instance we refer to the space required to encode the instance as an input to the algorithm. Hence, we determine the time an algorithm requires to solve a decision problem as a function of the input instance's size. For example, given a decision problem and an algorithm solving it, if the size of an instance is denoted by $n$ and we know the algorithm requires at most three times the size of an instance plus a fixed number of 42 computation steps, we can specify the number of computation steps required by the algorithm to solve an instance of the problem as $3n + 42$. Of course, there could also be an algorithm that requires only $3n + 10$ steps to solve the same problem. These, usually implementation dependent, differences can be neglected from an asymptotic perspective with regards to a growing instance size, such that we specify the number of computation steps required in "big omicron" (or more common "big O") notation, see the letter by Knuth [97]. Following this approach, both previously mentioned runtimes can be interpreted as equal since they both belong to $\mathcal{O}(n)$.

[1] The meanings of the terms "graph" and "vertex cover" are introduced in the next section.

[2] From here on, we use the terms Turing machine and algorithm interchangeably since, assuming the Church-Turing thesis, every feasible algorithm can be transformed into a Turing machine.

One further important aspect of computational complexity theory is that there are two different types of Turing machines, deterministic and nondeterministic ones. A *deterministic Turing machine* executes one computation step after the other, making a decision what step to execute next based on the available choices at every step. Imagining all possible computation paths of a computation as a large tree, a deterministic Turing machine follows a single path through the tree. Once the computation has ended, the machine either returns yes or no as an answer to the initial input instance. Of course, a deterministic machine could enter an infinite loop, never returning any result. However, in this case we cannot specify any runtime complexity, such that we assume that these computations always terminate. Contrarily, a *nondeterministic Turing machine* can follow multiple computation paths at the same time, basically exploring the whole computation tree in parallel. Thereby, a nondeterministic computation tree is allowed to contain infinite paths, such that not all computation paths of a nondeterministic computation tree must end. Therefore, a nondeterministic Turing machine does not decide for a given input instance whether it is a YES- or NO-instance, but only *accepts* YES-instances. To do so, a nondeterministic Turing machine returns yes as soon as it has found at least one computation path in the computation tree that results in a positive answer. Figure 1 illustrates this difference between deterministic and nondeterministic approaches in a graphical way for a simple computation tree. At this point we want to highlight that although nondeterministic Turing machines sound more preferable than deterministic ones since they can check multiple possibilities at once, so far no nondeterministic computer exists in the real world. Hence, in the real world every nondeterministic Turing machine must be simulated by a classical (deterministic) computer by sequentially processing all possible computation paths, losing the powerful advantage. Nevertheless, from a theoretical point of view, nondeterministic Turing machines are a fundamentally important concept of computational complexity theory.

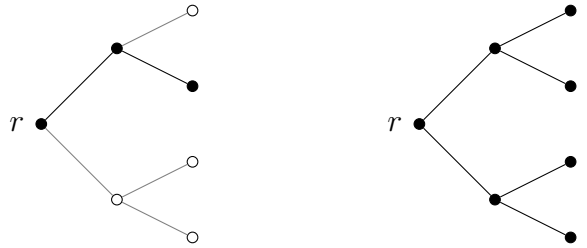



Figure 1: The computation path (filled, black) of a deterministic Turing machine (left) compared with the approach of a nondeterministic Turing machine (right). Both computations begin in the root $r$ of the tree and every vertex represents a computation step. Potential computation paths through the tree are connected via edges.

The *worst-case runtimes* of algorithms are specified with respect to the size $n$ of the input instance. For a deterministic algorithm it is defined as the maximum number of computation steps the algorithm requires to solve any instance of size $n$. Contrarily, for a nondeterministic algorithm it is defined as the maximum length of

a shortest, accepting computation path in the computation tree for any instance of size $n$.

Having provided a general overview of the fundamental aspects of computational complexity theory, next we define several important computational complexity classes that we require throughout our remaining work. A *complexity class* is a set that contains decision problems of equal complexity. One of the most prominent complexity classes is the class of problems that can be solved in *deterministic polynomial time*. We denote this class by P and a problem $\Psi$ belongs to P, i.e., $\Psi \in \mathrm{P}$, if there is a deterministic algorithm that can solve all instances of $\Psi$ in time polynomial in the input instance's size. We stick to the generally accepted terminologies and use the words "efficient" and "tractable" whenever some problem or function can be computed deterministically in time polynomial in the size of its input. Furthermore, there is also a class for problems that can be accepted in *nondeterministic polynomial time*. We denote this class by NP and a problem $\Gamma$ belongs to NP, i.e., $\Gamma \in \mathrm{NP}$, if there is a nondeterministic algorithm accepting all YES-instances of $\Gamma$ in polynomial time with respect to the input size. Lastly, there is the complexity class of coNP which contains the complements of the problems that belong to NP. In other words, if $I$ is an instance of $\Gamma$, then $I$ belongs to the complement of $\Gamma$ if and only if $I \notin \Gamma$ holds. Since every deterministic polynomial-time algorithm can also be interpreted as a nondeterministic polynomial-time algorithm, $\mathrm{P} \subseteq \mathrm{NP}$ follows immediately. However, the opposite inclusion, i.e., $\mathrm{NP} \subseteq \mathrm{P}$, is one of the most important, unresolved questions in computer science [43].

Before we can continue presenting further complexity classes required in our work, we need to introduce one additional concept from computational complexity theory, namely the concept of oracles. A Turing machine that is equipped with an *oracle* has access to some mechanism, whose inner workings are unknown, that is able to give yes or no answers to instances of a given decision problem within a single computation step. Such an oracle is a powerful addition for a Turing machine as the oracle can be used during the machine's computation to query an answer to some sub-instance within a single computation step that could help decide the initial instance. This concept of oracles can also be extended to complexity classes. For example, decision problems that can be solved in polynomial time by a deterministic Turing machine with access to an NP oracle, belong to the complexity class $\mathrm{P}^{\mathrm{NP}}$.

The subsequent complexity classes belong to the polynomial-time hierarchy PH introduced by Stockmeyer [133] in 1977. Thereby, the polynomial-time hierarchy is a hierarchy of complexity classes with increasing complexity that is recursively defined. The ground level of the hierarchy is defined as $\Sigma_0^{\mathrm{p}} = \Pi_0^{\mathrm{p}} = \Delta_0^{\mathrm{p}} = \mathrm{P}$ and the first level of the hierarchy consists of the complexity classes $\Sigma_1^{\mathrm{p}} = \mathrm{NP}$ and its complement $\Pi_1^{\mathrm{p}} = \mathrm{coNP}$. The $i$-th level of the polynomial-time hierarchy for $i \geq 2$, $i \in \mathbb{N}$, consists of $\Sigma_i^{\mathrm{p}} = \mathrm{NP}^{\Sigma_{i-1}^{\mathrm{p}}}$, $\Pi_i^{\mathrm{p}} = \mathrm{co}\Sigma_i^{\mathrm{p}}$, and $\Delta_i^{\mathrm{p}} = \mathrm{P}^{\Sigma_{i-1}^{\mathrm{p}}}$. Having all individual levels introduced, the complete polynomial-time hierarchy PH is defined as $\mathrm{PH} = \bigcup_{i \in \mathbb{N}} \Sigma_i^{\mathrm{p}}$. In later chapters we use the second level of the polynomial-time hierarchy, composed of $\Sigma_2^{\mathrm{p}} = \mathrm{NP}^{\mathrm{NP}}$, $\Pi_2^{\mathrm{p}} = \mathrm{coNP}^{\mathrm{NP}}$, as well as $\Delta_2^{\mathrm{p}} = \mathrm{P}^{\mathrm{NP}}$.

Besides these classes, we also require the class $\Theta_2^{\mathrm{p}} = \mathrm{P}^{\mathrm{NP[log]}}$, introduced by Papadimitriou and Zachos [121] in 1983. $\Theta_2^{\mathrm{p}}$ contains problems that can be solved by a deterministic polynomial-time Turing machine with access to some NP oracle which is queried at most logarithmically many times in the size of the input.

Having defined thus many complexity classes, natural questions arise about the relations of these classes among each other, say, which classes are contained in other classes, etc. This area of research is a wide subfield of complexity theory called structural complexity theory [34, 36, 105]. A lot of work has been done in this field, for example, it is a well-known result (see, e.g., [126]) that for every $i \in \mathbb{N}$, we have

$$\Sigma_i^{\mathrm{p}} \cup \Pi_i^{\mathrm{p}} \subseteq \Delta_{i+1}^{\mathrm{p}} \subseteq \Sigma_{i+1}^{\mathrm{p}} \cap \Pi_{i+1}^{\mathrm{p}}.$$

Besides such rather straightforward results there are also famous, way more involved results, such as the Karp-Lipton theorem, see Karp and Lipton [92], which states that if NP is a subset of $\mathrm{P}/_{\mathrm{poly}}$, then the polynomial-time hierarchy PH collapses to its second level.[3]

In order to prove that some problem $\Psi$ is a member of some complexity class $\mathcal{C}$, we must formulate an algorithm running within the specified computation time limits of $\mathcal{C}$, answering all instances of $\Psi$ correctly. If we are able to provide such a correctly working algorithm, it follows that $\Psi \in \mathcal{C}$ holds and doing so, we basically provide an upper bound for the computational complexity of the problem, as we know for sure that $\Psi$ can be solved within the means of $\mathcal{C}$. However, it might be that there exists a more efficient algorithm to solve $\Psi$ within the means of a complexity class of lower computational complexity. For two decision problems $\Psi$ and $\Gamma$, we can also try to translate instances of one problem into instances of the other problem. Therefore, we define a total function $f$ computable in time polynomial in its input that takes instances of $\Psi$ as input and returns instances of $\Gamma$ as output. Such a function $f$ is called a *polynomial-time many-one reduction* from $\Psi$ to $\Gamma$, if an instance $I$ is a YES-instance for $\Psi$ if and only if $f(I)$ is a YES-instance for $\Gamma$. In case such a function $f$ exists for two problems $\Psi$ and $\Gamma$, we also write $\Psi \leq_m^{\mathrm{p}} \Gamma$ and say that $\Psi$ is polynomial-time many-one reducible to $\Gamma$. Doing so, we show that $\Gamma$ is at least as computational complex as $\Psi$, as otherwise, we could translate instances of $\Psi$ into instances of the simpler problem $\Gamma$ and then solve these simpler instances. With the equivalence property of the reduction we could then deduce whether the original $\Psi$ instance is a YES-instance. A problem $\Psi$ is defined to be *$\mathcal{C}$-hard*, if for all $\Sigma \in \mathcal{C}$ it holds that $\Sigma \leq_m^{\mathrm{p}} \Psi$. Proving that a problem $\Psi$ is hard for some complexity class $\mathcal{C}$ is a way of showing a lower bound for the problem's complexity. That is because we show that $\Psi$ is at least as hard as every other problem in $\mathcal{C}$. Finally, a problem is defined to be *$\mathcal{C}$-complete* if it belongs to the complexity class $\mathcal{C}$ and is $\mathcal{C}$-hard. A common approach to prove that a decision problem $\Gamma$ is $\mathcal{C}$-hard consists of a polynomial-time many-one reduction from a $\mathcal{C}$-hard problem to $\Gamma$. This approach exploits the fact that $\leq_m^{\mathrm{p}}$ is a transitive relation, see [126].

[3] The complexity class $\mathrm{P}/_{\mathrm{poly}}$ contains all problems that can be decided via a family of polynomial-sized boolean circuits. For an introduction to boolean circuits and a formal definition of this complexity class we refer to the work by Arora and Barak [2].

One of the most known and thoroughly studied decision problems is Satisfiability, which is defined as follows [71].

| Satisfiability (SAT) | |
|---|---|
| **Given:** | A set of boolean variables $X$ and a boolean formula $\alpha$ over $X$ in conjunctive normal form. |
| **Question:** | Is there a boolean assignment for the variables in $X$ such that $\alpha$ evaluates to true? |

The subsequent example provides a Satisfiability instance and explains whether this instance is a yes- or no-instance.

**Example 2.1.** Let $(X, \alpha)$ with $X = \{x_1, x_2, x_3\}$ and

$$\alpha = (x_1 \vee \neg x_2 \vee x_3) \wedge (\neg x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_2 \vee \neg x_3)$$

be a Satisfiability instance. Setting $x_1 = x_2 = x_3 = \text{false}$ we found an assignment for the boolean variables in $X$ such that $\alpha$ evaluates to true. Consequently, $(X, \alpha)$ is a yes-instance for Satisfiability.

Besides being one of the most natural decision problems in computational complexity theory, this problem is also the first problem for which NP-completeness was proven. This result is due to Cook [42], who proved NP-completeness in 1971, and Levin [108], who proved NP-completeness independently in 1973. Since then, this result is known as the "Cook-Levin theorem" or simply as "Cook's theorem". Many more NP-completeness results have been proven up to today for an ever increasing number of decision problems. The book by Garey and Johnson [71] provides a noteworthy list of further, fundamental decision problems and their corresponding NP-completeness proofs.

Besides complete problems for the complexity class NP, there are uncountably many more complete problems for all sorts of complexity classes. For example, Schaefer and Umans [129, 130] provide a list of problems that are known to be complete for individual levels of the polynomial-time hierarchy. Listing these classes together with their complete problems is significantly beyond the scope of our work. Nonetheless, we want to introduce two further problems of which we make use in Chapter 3 of our work. The first problem is called Odd Max Satisfiability and was defined by Krentel [100].

| Odd Max Satisfiability (OMS) | |
|---|---|
| **Given:** | A set $X = \{x_1, \ldots, x_n\}$ of boolean variables and a boolean formula $\alpha$ over $X$. |
| **Question:** | Is $\alpha$ satisfiable and $x_n = 1$ in $\alpha$'s lexicographically maximum satisfying assignment $x_1 \ldots x_n \in \{0,1\}^n$? |

It was also Krentel who proved $\Delta_2^{\mathrm{p}}$-completeness for Odd Max Satisfiability in the same work in 1988. In a later chapter of our work we provide a polynomial-

time many-one reduction from OMS to a certain problem in order to show $\Delta_2^{\mathrm{p}}$-hardness for the latter problem. The second problem we introduce explicitly is called Succinct Set Cover and is defined as follows.

| Succinct Set Cover (SSC) | |
|---|---|
| **Given:** | A collection $S = \{\varphi_1, \ldots, \varphi_m\}$ of 3-DNF formulas on $n$ variables and an integer $k$. |
| **Question:** | Is there a subset $S' \subseteq S$ of size at most $k$ for which $\bigvee_{\varphi \in S'} \varphi \equiv \text{TRUE}$? |

This problem was defined and proven to be $\Sigma_2^{\mathrm{p}}$-hard by Umans [137] in 1999. Similarly to the previous problem, we use this problem to prove $\Sigma_2^{\mathrm{p}}$-completeness for a newly defined problem.

The last technique related to the field of computational complexity theory that we introduce in this section is the technique of *linear programming*. Initially, this technique originates from the area of mathematics and is used to calculate optimal solutions to linear optimization problems. Transferring this approach to computational complexity, we subsequently formulate a feasibility version of this technique as decision problem [71].

| Linear Programming (LP) | |
|---|---|
| **Given:** | An integer matrix $A \in \mathbb{N}^{m \times n}$, integer vectors $d \in \mathbb{N}^m$ and $c \in \mathbb{N}^n$, and an integer $B \in \mathbb{N}$. |
| **Question:** | Is there a rational vector $x \in \mathbb{Q}^n$ such that $A \cdot x \leq d$ holds line-wise and $c^T \cdot x \geq B$ is true? |

Hačijan [82] proved that this version of Linear Programming, as defined above, can be solved in P. Contrarily, if one demands the vector $x$ to consist of integers instead of rational numbers, i.e., $x \in \mathbb{Z}^n$, one obtains the Integer Linear Programming problem, which is known to be NP-complete, see the work by Karp [91].[4] A lot of different problems from various areas can be encoded as linear programs. For example, Fitzsimmons and Hemaspaandra [65] use the approach to encode election-related problems as linear programs. In a later chapter of our work we follow this approach as well when encoding certain decision problem instances as linear program instances.

With the definition of this technique we end our introduction to the field of computational complexity theory. We covered all concepts required for later parts of our work while not diving too much into narrow details. Besides the books mentioned at the beginning of this section, we also recommend the works by Tovey [134] and Hemaspaandra and Ogihara [89] for further primers towards this research area.

[4] Actually, Karp proved NP-completeness for the 0-1-Integer Programming problem, a special case of the general variant.

## 2.2 Graph Theory

Having introduced all required concepts of computational complexity, this section is devoted to the research area of graph theory. First, we define the general concepts of graphs as well as important, related terms and notations. Afterwards, we define several different types of graph parameters and classes of graphs. Last, we discuss the concepts of stability and unfrozenness, introduce related definitions, and define corresponding decision problems.

In graph theory it is commonly accepted that Euler [61] with his work on the Königsberg Bridge Problem in 1741 "became the father of graph theory" as Harary [83] formulates it. The Königsberg Bridge Problem relates to the city of Königsberg, which was build on two sides of a river with two islands in the middle of the river. The city was connected across the river via seven bridges and the Königsberg Bridge Problem asked whether it is possible to walk once around the complete city while using every bridge exactly one time. In the previously mentioned work, Euler proved that it is impossible to find a route through the city visiting every bridge exactly once. To do so, Euler generalized the map of Königsberg in a graph and solved the problem for the resulting graph, see Figure 2 for an illustration. Since the formal concept of

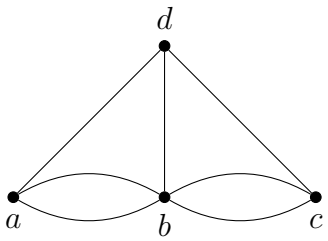



Figure 2: An illustration of the Königsberg Bridge Problem as a graph. Every vertex $a, b, c, d$ abstracts some part of the city Königsberg and every edge between two vertices describes a bridge across the river. Vertices $a$ and $c$ represent parts of the mainland while vertices $b$ and $d$ represent islands.

a graph had not yet been established at this time, Euler's work laid the foundation for what we call graph theory today. The research area of graph theory is extremely diversified and tightly interconnects with various research areas of other disciplines. In the more recent research history of graph theory, the book by Harary [83] is one of the essential works providing a proper introduction to graph theory. Besides this work, we also refer the reader to the more modern works on graph theory by West [142] and Bollobás [27] for an exhaustive introduction.

Following the general definition, a *graph*, denoted by $G = (V, E)$, consists of a set of *vertices* $V$ and a set of *edges* $E \subseteq \{\{u, v\} \mid u, v \in V\}$ between those vertices. In our work, we only work with *finite graphs*, i.e., we always implicitly assume that $|V| < \infty$ is true. However, there is also the subject of infinite graph theory [47]. The subsequent example formalizes the graph depicted in Figure 2.

**Example 2.2.** The graph $G = (V, E)$ used to abstract the city of Königsberg with its two islands and seven bridges possesses the four vertices $V = \{a, b, c, d\}$ and the seven edges $E = \{\{a, b\}, \{a, b\}, \{b, c\}, \{b, c\}, \{a, d\}, \{b, d\}, \{c, d\}\}$.

With this example it becomes clear that both notations of a graph, i.e., illustrated as in Figure 2 and formally via sets as in the previous example, are equally expressive. Depending on the use case, one of both notations might be more advantageous and throughout our work we use both notations as it suits best.

Furthermore, a graph that allows multiple edges between two vertices is called a *multigraph* and a graph that only allows at most one edge between to vertices is called a *simple graph*. Given a graph $G$, we sometimes denote its set of vertices by $V(G)$ and its set of edges by $E(G)$. A graph, be it a multigraph or a simple graph, can be undirected or directed. So far, we have only defined *undirected graphs*. A *directed graph* $G = (V, E)$, also abbreviated as *digraph*, consists of a set of vertices $V$, and a set of directed edges $E \subseteq V \times V$ between its vertices. A directed edge, sometimes also called an *arc*, from vertex $u$ towards vertex $v$ is denoted by $(u, v)$. In our remaining work we restrict ourselves to undirected, simple graphs and hence, whenever we write graph, we refer to an undirected, simple graph.

As mentioned earlier and as we see in later parts of this work, the subject of graph theory is closely related to many other fields of research within and beyond computer science. Thus, it comes at no surprise that there exists a large variety of further concepts building on graphs. Subsequently, we introduce some of these concepts that we require in ensuing chapters of our work. Given a graph $G$, we call

1. a subset of vertices $V' \subseteq V(G)$, such that for every two vertices $u, v \in V'$ it holds that $\{u, v\} \notin E(G)$, an *independent set*,

2. a subset of vertices $V'' \subseteq V(G)$, such that for every two vertices $u, v \in V''$ it holds that $\{u, v\} \in E(G)$, a *clique*,

3. a subset of vertices $V''' \subseteq V(G)$, such that for every edge $\{u, v\} \in E(G)$ it holds that $u \in V'''$ or $v \in V'''$, a *vertex cover* for $G$, and

4. a function $c\colon V(G) \to \mathbb{N}$, assigning every vertex of $G$ a nonnegative integer such that for every edge $\{u, v\} \in E(G)$ it holds that $c(u) \neq c(v)$, a *legal coloring* of $G$.[5]

The subsequent example applies the previously introduced concepts to the graph from Figure 2.

**Example 2.3.** For the multigraph $G$ used to abstract the city of Königsberg, the vertices $I = \{a, c\}$ form an independent set, the vertices $C = \{a, b, d\}$ build a clique and the vertices $V' = \{b, d\}$ provide a vertex cover. Assigning the colors

[5]The term coloring is also used with respect to applications related to coloring maps, for example, see the work by Appel and Haken [1] for more in-depth background.

$c(a) = c(c) = 1$, $c(b) = 2$, and $c(d) = 3$, we obtain a legal coloring $c$ of $G$ with three colors.

Based on these special subsets of vertices, next we introduce the notion of a graph parameter. To introduce the concept of a graph parameter, let us denote the set of all undirected, simple graphs by $\mathcal{G}$. Then, a *graph parameter* is a function $\psi\colon \mathcal{G} \to \mathbb{N}$ that assigns every simple graph $G \in \mathcal{G}$ a nonnegative integer in $\mathbb{N}$. Of course, the concept of a graph parameter can also be extended to multigraphs as well as digraphs [78]. Subsequently, we introduce four of the most known and well-studied graph parameters which we use in Chapter 6. Given an undirected, simple graph $G \in \mathcal{G}$,

1. the *independent set number* $\alpha(G)$ describes the size of a largest independent set in $G$,

2. the *clique number* $\omega(G)$ describes the size of a largest clique in $G$,

3. the *vertex cover number* $\beta(G)$ describes the size of a smallest vertex cover for $G$, and

4. the *chromatic number* $\chi(G)$ describes the minimum number of colors, such that a legal coloring for $G$ using this number of colors exists.

The subsequent example provides values for all four previously established graph parameters and a modified version of the graph from Figure 2.

**Example 2.4.** Since we defined graph parameters only for simple graphs and not for multigraphs, let us denote by $G'$ the graph one obtains when removing one of the two edges between $a$ and $b$ as well as $b$ and $c$. The resulting graph $G'$ is no longer a multigraph but a simple graph. Despite these modifications, all sets mentioned in Example 2.3 still possess their special meanings. Furthermore, all the sets we listed in the previous example are optimal in the sense that they are maximal for a possible independent set and a clique and respectively minimal for a vertex cover and some coloring. Hence, the subsequent values

$$\alpha(G') = 2,\ \omega(G') = 3,\ \beta(G') = 2,\ \text{and}\ \chi(G') = 3$$

for all four graph parameters follow immediately.

Besides determining values of specific graph parameters for given graphs, there has also been a variety of research on the relations of graph parameters [76, 138, 144]. One of the well-known relationships for two graph parameters stems from the theorem by Gallai [68], also known as "Gallai's theorem", which we apply in later parts of our work, too.

**Theorem 2.5** (Gallai's theorem)**.** Let $G \in \mathcal{G}$ be a graph. Then it holds that $|V(G)| = \alpha(G) + \beta(G)$ is true.

Furthermore, graph parameters are also used to characterize graph classes [79, 77, 26]. Thereby, a *graph class* $\mathcal{C} \subseteq \mathcal{G}$ is a subset of $\mathcal{G}$ consisting of graphs that satisfy a certain property. Subsequently, we introduce graph classes which we require in later parts of our work. To begin, let us define the class of trees, denoted by $\mathcal{T}$. A graph $G \in \mathcal{G}$ is a *tree*, i.e., $G \in \mathcal{T}$, if $G$ is connected and does not possess cycles of length greater than or equal to three. Thereby, a graph is *connected* if, starting at an arbitrary vertex and following the graph's edges, all other vertices in the graph can be reached. A graph $G$ possesses a *cycle of length* $n$, if there exists a sequence of $n$ adjacent[6] edges in $G$ that starts and ends at the same vertex without visiting any vertex in between multiple times. Having defined the class of trees, we can immediately introduce the next class, namely forests, denoted by $\mathcal{F}$. A graph $G \in \mathcal{G}$ is defined to be a *forest*, i.e., $G \in \mathcal{F}$, if $G$ can be written as the union of several trees. Hence, while a tree needs to be a connected graph, i.e., there is always a path between any two vertices of a tree, a forest can consist of several disconnected trees. Next, we introduce the class of bipartite graphs $\mathcal{B}$. A graph $G$ is defined to be a *bipartite graph*, if its vertices can be partitioned into two disjunct sets $V_1, V_2 \subseteq V(G)$, such that every edge $e \in E(G)$ is incident[7] to exactly one vertex from $V_1$ and one vertex from $V_2$. While the previous graph classes are widely known and, for example, also introduced in the works by Harary [83] and West [142], the class of co-graphs, which we introduce ensuingly, is less known. The first graph theoretic definition of co-graphs was given by Lerchs [106, 107] in 1971 and 1972, although, according to Corneil, Lerchs, and Burlingham [45], co-graphs already emerged independently under different names earlier in several areas of mathematics. The subsequent definition is a slightly modified version of the one given by Corneil, Lerchs, and Burlingham [45]. To define co-graphs, we require the definitions of two operations on graphs. First, for two vertex-disjoint graphs $G, H \in \mathcal{G}$, we define by $\cup\colon \mathcal{G} \times \mathcal{G} \to \mathcal{G}$ the *disjoint union* of two graphs as

$$G \cup H \mapsto (V(G) \cup V(H), E(G) \cup E(H)).$$

In plain terms, we unite the vertex sets of both graphs as well as the edge sets of both graph as previously done when uniting trees to a forest. Second, we define by $+\colon \mathcal{G} \times \mathcal{G} \to \mathcal{G}$ the *join* of two vertex-disjoint graphs as

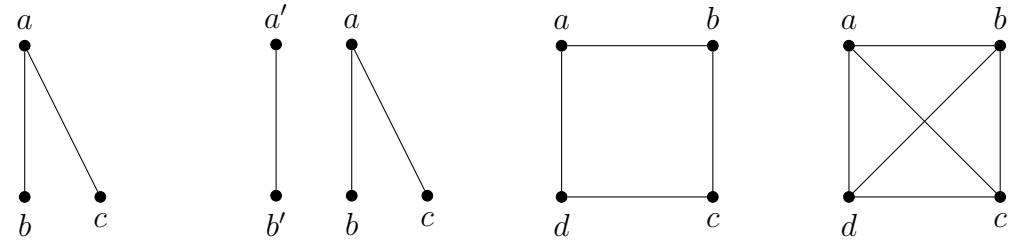
$$G + H \mapsto (V(G) \cup V(H), E(G) \cup E(H) \cup \{\{u, v\} \mid u \in V(G), v \in V(H)\}).$$

In other words, the join of two graphs unites their vertex as well as edge sets and adds edges between all vertices of the first graph and all vertices of the second graph. With these two definitions at hand we can now define the class of *co-graphs*, denoted by $\mathcal{C}$, recursively as follows. Graph $G = (\{v\}, \emptyset)$ is a co-graph and if $G_1$ and $G_2$ are two co-graphs, then $G_1 \cup G_2$ and $G_1 + G_2$ are co-graphs, too.

Having all these different graph classes introduced, the next example provides exemplary graphs for all these classes and highlights some of their relations.

[6] Two edges are said to be *adjacent* if they share a common vertex and two vertices are said to be *adjacent* if they share a common edge.

[7] An edge $e$ and a vertex $v$ are *incident* if $v \in e$ holds.




(a) $G_1$ (b) $G_2$ (c) $G_3$ (d) $G_4$

Figure 3: Exemplary graphs for the graph classes of trees, forests, bipartite graphs, and co-graphs.

**Example 2.6.** The graph $G_1$ in Figure 3a belongs to the class of trees. It is easy to see that $G_1$ is connected and contains no cycle of length greater than or equal to three, such that $G_1 \in \mathcal{T}$ holds. Graph $G_2$ in Figure 3b contains no cycle of length greater than or equal to three but is not connected, such that $G_2 \in \mathcal{F}$ is a forest but not a tree. Solely based on the definitions of these two graph classes it immediately follows that every tree is a forest but not every forest is a tree, hence $\mathcal{T} \subseteq \mathcal{F}$ holds. Graph $G_3$ in Figure 3c is a bipartite graph. We can partition its vertices $V(G_3) = \{a, b, c, d\}$ into $V_1 = \{a, c\}$ and $V_2 = \{b, d\}$. Then, all edges are between a vertex from $V_1$ and a vertex from $V_2$ such that $G_3 \in \mathcal{B}$ follows. Every tree is a bipartite graph since we can assign the vertices of a tree in alternating sequence to $V_1$ and $V_2$, such that $\mathcal{T} \subseteq \mathcal{B}$ follows. Additionally, since a forest consists of several trees and as a bipartite graph must not be connected, it also follows that $\mathcal{F} \subseteq \mathcal{B}$ holds. Lastly, graph $G_4$ from Figure 3d is a co-graph. To see this, we must build $G_4$ via the previously defined operations out of smaller graphs. We know that the graph consisting of a single vertex without any edges is a co-graph. Let us denote this graph simply by its vertex $v$. Then, we obtain $G_4$ by $G_4 = a + b + c + d$, i.e., we execute the join operation three times in sequence, each time joining the graph obtained from the previous operation with a new graph consisting of a single vertex.

At this point we finish our general introduction of graph theory and narrow down the focus towards the concept of stability of graphs. The original idea for the concept of stability of graphs was introduced by Dirac [51] in 1952. Since then, a substantial amount of research has been done in this area [13, 75, 90]. Subsequently, we present the general, underlying idea of stability of graphs, following the notations by Frei, Hemaspaandra, and Rothe [67]. Let $\xi$ be a graph parameter and $G \in \mathcal{G}$ a simple graph. $G$ is called *$\xi$-stable* if the removal of any single edge from $G$ does not alter the value of $\xi(G)$. Contrarily, if there is an edge $e \in E(G)$ such that $\xi(G) \neq \xi(G-e)$ holds, $G$ is said to be *$\xi$-critical*.[8] This concept can also be extended to vertices. A graph $G \in \mathcal{G}$ is said to be *$\xi$-vertex-stable* if the removal of any single vertex from $G$ does not alter the value of $\xi(G)$. Analogously, if there is a vertex $v \in V(G)$, such that $\xi(G) \neq \xi(G - v)$ holds, $G$ is called *$\xi$-vertex-critical*. So far, both terms,

[8] For a graph $G$ and an edge $e \in E(G)$ (a vertex $v \in V(G)$) we denote by $G-e$ (respectively, $G-v$) the graph one obtains by removing $e$ (respectively, $v$) from $G$.

$\xi$-stability and $\xi$-vertex-stability, are related to removing an edge or a vertex from a graph. Besides this approach, there is also the concept of unfrozenness where one does not remove but add new edges or vertices together with incident edges to a graph. The corresponding terminology is *$\xi$-unfrozen* and *$\xi$-vertex-unfrozen* if a graph parameter's value does not change for a given graph and *$\xi$-frozen* and *$\xi$-vertex-frozen* in case the value changes.

Having defined these terms, one can as well define corresponding decision problems and approach this field of research from a computational complexity theoretic perspective. For example, Frei, Hemaspaandra, and Rothe [67] formulated the decision problem of $\xi$-Stability as follows.

| $\xi$-Stability | |
|---|---|
| **Given:** | A graph $G \in \mathcal{G}$. |
| **Question:** | Is $G$ $\xi$-stable? |

The decision problems for the remaining definitions, i.e., vertex-stability, unfrozenness, and vertex-unfrozenness are defined analogously. Up to the work by Frei, Hemaspaandra, and Rothe [67], the computational complexity study of stability of graphs was not very fruitful. It was only their work which initiated a "systematic study of stability of graphs in terms of their computational complexity", as the authors phrased it. Within their work, the authors established several $\Theta_2^{\mathrm{p}}$-completeness results for various of the earlier defined decision problems. We require these decision problems in Chapter 6 of our work, when we develop efficient algorithms to solve the same decision problems for special graph classes.

## 2.3 Computational Social Choice and Fair Division

In this section we introduce three different fields of research that we address by our results in the ensuing chapters. The first two fields of judgment aggregation and preference aggregation by voting stem from the area of COMSOC. The last field, cake-cutting, belongs to the area of fair division. Both areas, COMSOC and fair division, are closely related [127].

COMSOC has its origins in the research area of social choice theory, see, e.g., [32] for an exhaustive historical overview. Social choice theory addresses the aggregation of preferences or judgments of several individuals into collective outcomes. The book by Arrow [3] and the handbook by Arrow, Sen, and Suzumura [4] provide thorough introductions. COMSOC builds on this research and connects it to the field of computer science. For example, COMSOC addresses the practical feasibility of designed voting rules. Bartholdi III, Tovey, and Trick [12] were the first to study how hard, in terms of computational complexity, it is to determine the winner of an election for a given voting rule: While it is favorable to design a voting rule that satisfies every participating agent's preferences, this voting rule does not provide any value if its outcome cannot be calculated in reasonable time as its computational complexity is simply too high. Since then, COMSOC has evolved into a central

area of research with respect to multiagent systems and artificial intelligence [60]. It reaches across several sub fields such as preference aggregation by voting [22], control, manipulation, and bribery of (single-peaked) elections [86], and judgment aggregation [17].

While the area of COMSOC addresses questions with regards to preference aggregating rules, their properties, and their computational complexity, fair division addresses similar questions with regards to algorithms for the fair division of goods among agents [5]. Though familiar, the perspective shifts with respect to fair division of goods. In COMSOC, agents are interested in the general collective outcome that holds for everyone. Contrarily, in fair division, agents are only interested in the share of the good(s) allocated to them [103]. Thereby, fair division can be compartmentalized into two different settings, the fair division of arbitrarily divisible goods, named cake-cutting, and the fair division of indivisible goods, sometimes called multiagent resource allocation [38, 118]. In both settings, we are given a group of agents with preferences over a good or multiple goods that shall be shared among the agents. The only difference is that for cake-cutting the single good, usually metaphorically represented by a cake, can be cut into arbitrary pieces, while for fair division of indivisible goods we have a set of goods that cannot be divided into smaller parts [109, 103].

### 2.3.1 Judgment Aggregation

In this section we introduce the research area of judgment aggregation. The basic setting of judgment aggregation addresses the aggregation of multiple, individual judgments by a certain, predefined rule into one aggregated, collective outcome. Applications of judgment aggregation can be found across a wide variety of different subjects such as jurisdiction, politics, philosophy, economics, and computer science [53, 17, 37, 70]. For example, multiple judges in a court judging over a case need to aggregate their individual judgments into one collective judgment, autonomous vehicles need to make aggregated decisions how to behave when encountering each other on the street, or several communicating sensors in a network must agree on the detection of external influences.

To provide a formal introduction to judgment aggregation, we follow the notations and definitions by Endriss [53]. When aggregating judgments, we call the participating group of agents *judges* and denote them by $\{1, \dots, r\}$ for $r \in \mathbb{N}$. The judges provide individual judgments over an *agenda* $\Phi$ which consists of boolean formulas. The agenda $\Phi$ is assumed to be finite, *closed under complements* (for every $\varphi \in \Phi$, we demand $\neg\varphi \in \Phi$), *nontrivial* (there are $\varphi, \phi \in \Phi$ such that $\{\varphi, \phi\}, \{\varphi, \neg\phi\}, \{\neg\varphi, \phi\}$, and $\{\neg\varphi, \neg\phi\}$ are all satisfiable), and it does not contain tautologies or contradictions. In some cases the agenda is partitioned into two disjoint subsets $\Phi_+$ and $\Phi_-$, where for every $\varphi \in \Phi_+$ it must hold that $\neg\varphi \in \Phi_-$. An *individual judgment* is denoted by $J \subseteq \Phi$ and we say that $J$ is

1. *complete*, if for all $\varphi \in \Phi$ either $\varphi \in J$ or $\neg\varphi \in J$ holds,

2. *complement-free*, if for every $\varphi \in \Phi$ it holds that not both formulas $\varphi$ and $\neg\varphi$ are in $J$ at the same time, and

3. *consistent*, if there is a boolean assignment that satisfies all formulas in $J$ at once.

The set of all complete and consistent judgments over $\Phi$ is denoted by $\mathcal{J}(\Phi)$ and the list of all participating judges' individual judgments $P_1, \ldots, P_r$ over $\Phi$ is called the judges' *profile*, denoted by $P = (P_1, \ldots, P_r) \in \mathcal{J}(\Phi)^r$. Finally, a *judgment aggregation rule* is a function $R\colon \mathcal{J}(\Phi)^r \to 2^\Phi$ that maps a profile $P$ of $r$ individual judgments to an aggregated outcome. Thereby, we say that $R$ is

1. *anonymous*, if $R(P)$ is independent of the order of the individual judgments in $P$,

2. *neutral*, if it approaches every element of the agenda the same way, i.e., if two elements $\varphi_1, \varphi_2$ are approved by the same set of judgments either both elements or none belong to $R(P)$, and

3. *independent*, if adding an element $\varphi_1$ to the aggregated outcome only depends on the individual judgments for this element.

Having defined all the basic notions and terms, we want to emphasize that besides this introduced formal model of judgment aggregation, there are further formal models actively used in the area of judgment aggregation, e.g., binary aggregation [73, 54]. To become more familiar with the introduced model, let us take a look at the subsequent example.

**Example 2.7.** Assume we have the agenda $\Phi = \{a, b, c, \neg a, \neg b, \neg c\}$ and three judges $\{1, 2, 3\}$ with profile $P$ as described in Table 1. If the judges agree to apply the majority rule *Maj*, i.e., to add only elements from the agenda to the collective outcome that are supported by a majority of the judgments, they obtain as aggregated outcome $a, b, \neg c$.

| $P$ | $a$ | $b$ | $c$ |
|---|---|---|---|
| $P_1$ | TRUE | TRUE | TRUE |
| $P_2$ | TRUE | FALSE | FALSE |
| $P_3$ | FALSE | TRUE | FALSE |
| *Maj* | TRUE | TRUE | FALSE |

Table 1: Individual judgments and aggregated outcome for three judges applying the majority rule *Maj*. TRUE indicates that an element from the agenda is part of an individual judgment or the collective outcome and FALSE shows that the element's negation is part of it.

Following the previous example, let us assume that outer circumstances enforce the logical requirement $a \wedge b \Leftrightarrow c$ to always hold. Such a requirement is also known in judgment aggregation as *external doctrine* and could be used to report dependencies one needs to pay respect to when modeling a scenario within judgment aggregation. Doing so, we can see that all three individual judgments satisfy the external doctrine while the aggregated outcome does not. This observation, initially introduced in 1986 by Kornhauser and Sager [99] in the context of jurisdiction and known as doctrinal paradox, describes the case of a majority based judgment aggregation of three individual, consistent judgments into an inconsistent collective outcome. Pettit [122] generalized the implications of the doctrinal paradox as the discursive dilemma and explained that such paradoxical results can always occur when a group of individuals wants to aggregate their individual judgments over a set of related boolean elements.

With this example and the observations thereafter, an important question regarding the used judgment aggregation rule arises. It could be the case that the majority rule is not an optimal choice and there are more sophisticated rules preventing paradoxical situations like the previous one. Many different judgment aggregation rules have been invented and studied in the literature. For example, Dietrich and List [49] formalized the concept of so-called *quota rules*. A quota rule adds an element from the agenda to the collective outcome if the element reaches a predefined quota of acceptance among the individual judgments. The most known quota rule is probably the majority rule, which applies the same quota of 1/2 to all elements of the agenda. A different family of aggregation rules are *distance-based aggregation rules*. For these rules, the aggregated outcome is determined by choosing a complete and consistent judgment as outcome that reduces the distance towards the profile of individual judgments. For example, the distance could be calculated as some Hamming distance, yielding the generalized Kemeny rule [53, 93]. Furthermore, there is the concept of *premise-based aggregation rules*. For a premise-based aggregation rule one splits the agenda into premises and conclusions, applies some given rule to the premises of the agenda and afterwards, deduces the membership for the conclusions based on the aggregated premises [53]. List [110] introduced the idea of sequential judgment aggregation rules. A sequential judgment aggregation rule iterates in a predefined order over the agenda and applies some underlying judgment aggregation rule as base rule. In each round, for the next element of the agenda to be considered the decision is made whether to add it or its negation to the collective outcome. Thereby, one is either able to deduce from the already added elements of previous rounds whether the current one or its negation shall be added or, otherwise, applies the underlying rule to the individual judgments to decide if the element shall be added. An example for such a sequential judgment aggregation rule is the sequential majority rule, applying the majority rule as underlying base rule.

In 2002, List and Pettit [112] proved that not only the majority rule is prone to the discursive dilemma, but every judgment aggregation rule that satisfies a certain set of axioms, in particular, every judgment aggregation rule that satisfies the axioms of anonymity, neutrality, and independence [53]. Besides these three properties of

anonymity, neutrality, and independence, further properties for judgment aggregation rules, such as *monotonicity*, *complement-freeness*, *completeness*, *consistency*, *nondictatorship*, and *systematicity*, were defined [17]. Additionally, there are also properties for agendas, such as the *median property* which is satisfied by an agenda if every of its inconsistent subsets itself possesses an inconsistent subset containing at most two elements [114]. Based on all these properties for judgment aggregation rules and agendas, there are many works that have studied (im)possibility results with respect to these properties. In other words, which combinations of properties can be satisfied by an individual aggregation rule and which combinations are impossible [113, 53, 48, 115]. Beyond these (im)possibility results, it is also possible to formulate characterizations of judgment aggregation rules based on the properties they satisfy [49].

Next to these directions of research addressing the invention of new, advanced judgment aggregation rules and proving (im)possibility results, there is also a branch of computational complexity analysis in judgment aggregation. This research direction was initiated by Endriss, Grandi, and Porello [55] in 2012. In their work, the authors formulated two types of decision problems related to judgment aggregation. Let $R$ denote some judgment aggregation rule. The first problem the authors defined is the so-called $R$-Winner problem.

| $R$-Winner | |
|---|---|
| **Given:** | An agenda $\Phi$, a profile of individual judgments $P \in \mathcal{J}(\Phi)^r$, and an element $\varphi \in \Phi$. |
| **Question:** | Does $\varphi \in R(P)$ hold? |

The winner problem is used to determine how hard it is to calculate the collective outcome of a given profile under some judgment aggregation rule. Obviously, a judgment aggregation rule that, for example, always returns a consistent outcome but which cannot be calculated efficiently is of limited value in most practical contexts. Many works studying the computational complexity of the winner problem for various judgment aggregation rules have been published [104, 54, 56, 81, 57].

The second problem that was introduced is the $R$-hd-Manipulation problem. Initially, the ideas of manipulation and strategy-proofness were introduced to judgment aggregation by List [111] and Dietrich and List [50] in 2006 and 2007. Endriss, Grandi, and Porello [55] then approached these concepts from a computational complexity perspective and introduced a first manipulation decision problem related to hamming distances. The subsequent formulation of the manipulation problem is due to Baumeister, Erdélyi, Erdélyi, and Rothe [16].

| $R$-Manipulation | |
|---|---|
| **Given:** | An agenda $\Phi$, a profile of individual judgments $P \in \mathcal{J}(\Phi)^r$, and some subset $M \subseteq P_r \in P$ preferred by the manipulating judge $r$. |
| **Question:** | Is there some judgment $P'_r \in \mathcal{J}(\Phi)$ that the manipulator can submit to $P$ instead of his true preferences $P_r$ such that $M \subseteq R(P)$ holds? |

In the same work, the authors proved NP-completeness for this version of the manipulation decision problem for several quota-based rules and introduced the concept of bribery to judgment aggregation. In contrast to manipulation, where an individual judge states false preferences in order to manipulate the collective outcome towards his preferences, bribery addresses the question whether a limited number of judges can be bribed by some external briber to change their submitted judgments, so that the collective outcome shifts towards the briber's preferences.

| $R$-Bribery | |
|---|---|
| **Given:** | An agenda $\Phi$, a profile $P \in \mathcal{J}(\Phi)^r$, a consistent subset $B \subseteq J' \in \mathcal{J}(\Phi)$ preferred by the briber, and a positive integer $k \in \mathbb{N}$. |
| **Question:** | Is it possible to change at most $k$ individual judgments from $P$ such that for the modified profile $P'$ it holds that $B \subseteq R(P')$? |

For this problem, Baumeister, Erdélyi, Erdélyi, and Rothe [16] were able to prove various NP-completeness results related to premise-based aggregation rules. This branch of research within judgment aggregation tries to classify decision problems like the previously introduced ones in terms of computational complexity for different judgment aggregation rules. One part addresses the question how hard it is to determine the aggregated outcome for a given profile under some specified judgment aggregation rule. This is mostly done by determining the computational complexity of the winner problem for different judgment aggregation rules. For a practicably usable judgment aggregation rule one desires the winner problem to possess a low computational complexity, such that the aggregation of individual judgments into a collective outcome can be executed efficiently. Another objective is to determine whether certain judgment aggregation rules are protected against manipulation and bribery attempts via high computational complexity barriers. To prove or disprove such protection, one studies the computational complexity of manipulation- and bribery-related problems. Hence, for these problems one hopes for high computational complexities such that manipulation of the collective outcome by an individual judge as well as by a briber bribing several judges is computationally infeasible.

In Chapter 3 of our work we study the computational complexity of the winner problem as well as of manipulation-related decision problems for sequential judgment aggregation rules with a special focus on quota rules as underlying rules.

### 2.3.2 Preference Aggregation by Voting

In this section we introduce the field of preference aggregation by voting with a focus on the computational complexity perspective. Our basic notations and definitions

follow the work by Baumeister and Rothe [22]. The general setting is as follows: Given a set of several candidates, a group of voters express their preferences over these candidates. A voting rule is applied to aggregate the votes over the candidates and determine one or multiple winners as an outcome of the election. More formally, an *election* $(C, V)$ consists of a finite set of *candidates* $C$ and a finite list of *votes* $V$ over the candidates in $C$.[9] A *vote* $v \in V$ represents a strict, linear order over the candidates in $C$ and is used to express the preferences over the candidates in $C$ by a voter participating in the election. To denote a vote $v$, we use $>$ as a relation over the candidates in $C$. For example, when we have the candidates $C = \{a, b, c, d\}$ and a voter prefers candidate $b$ over $a$, $a$ over $d$, and $d$ over $c$, we denote the vote $v$ by

$$b > a > d > c.$$

Given a set of candidates $C$, we assume the relation $>$ to be

1. *connected*, i.e., for every two candidates $a, b \in C$ either $a > b$ or $b > a$ must hold,

2. *transitive*, i.e., for every three candidates $a, b, c \in C$ it holds that if $a > b$ and $b > c$ is true, then $a > c$ must hold, and

3. *asymmetric*, i.e., for every two candidates $a, b \in C$ it holds that if $a > b$ is true, $b > a$ cannot hold.

A vote $v \in V$ in which all candidates from $C$ occur at some position in the preference order is called a *total preference* or *total vote* over $C$. Contrarily, a vote that is missing some candidates from $C$ in its preference order is called a *partial preference* or *partial vote* over $C$.

Given an election $(C, V)$ we want to determine the winners of the election by some *voting rule* $\mathcal{E}$. A voting rule $\mathcal{E}$ takes as input an election $(C, V)$ and outputs a subset $\mathcal{E}(C, V) \subseteq C$ of the candidates as winners of the election. A candidate $c \in \mathcal{E}(C, V)$ that is the only winner of an election $(C, V)$, i.e., when $\mathcal{E}(C, V) = \{c\}$ holds, is called a *unique winner*. A candidate $c'$ that is a winner of an election together with other candidates is called a *nonunique winner*, i.e., when $\{c'\} \subsetneq \mathcal{E}(C, V)$.

There are two different ways how to represent an election. An election in *normal representation* is specified as previously described. Alternatively, an election can also be specified in *succinct representation*. In this case, we write the list of votes $V$ in a different way. Instead of adding identical votes multiple times to $V$, identical votes are only added once to $V$ together with a number indicating the multiplicity of this vote. These distinct forms of representation do not have any direct impact on the election or its outcome but differ in size, i.e., the space one requires to write down an election in these representations is different. These changes in size can affect complexity bounds of potential algorithms, see the work by Fitzsimmons and Hemaspaandra [65] for a thorough study of the implications.

[9] It is important to emphasize that while $C$ is a set of candidates, $V$ is a list (or multiset) of votes over $C$ as multiple voters with identical preferences can participate in an election.

Having defined the foundations of elections, candidates, and votes, next, let us turn to several voting rules. To begin, we define the concept of a *scoring protocol*. A scoring protocol possesses a *scoring vector* $\alpha$ that describes how many points a candidate obtains for her position in a vote. Given an election $(C, V)$ with $n$ candidates, a scoring vector

$$\alpha = (\alpha_1, \ldots, \alpha_n)$$

consists out of $n$ nonnegative integers $\alpha_i$ with $\alpha_i \geq \alpha_{i+1}$ for $1 \leq i \leq n$. A candidate $c \in C$ obtains $\alpha_i$ points for every vote $v \in V$ where she takes the $i$-th position and $c$'s overall score in the election corresponds to the sum of her points over all votes. Finally, all candidates with the highest score win the election. There a several scoring protocols with specific names that we study in our work:

1. *$k$-approval*: The $k$-approval scoring protocol possesses the following scoring vector

   $$\alpha = (\underbrace{1, \ldots, 1}_{k}, 0, \ldots, 0),$$

   where the first $k$ entries are ones and the remaining $n - k$ entries for $n$ candidates are zeros. 1-approval is also known as *plurality*.

2. *$k$-veto*: The $k$-veto scoring protocol possesses the following scoring vector

   $$\alpha = (1, \ldots, 1, \underbrace{0, \ldots, 0}_{k}),$$

   where the first $n - k$ entries are ones and the last $k$ entries are zeros for $n$ candidates. 1-veto is also known as *veto*.

3. *Borda Count*: The Borda Count scoring protocol possesses the following scoring vector

   $$\alpha = (n - 1, n - 2, \ldots, 1, 0)$$

   for $n$ candidates and is often only referred to as the Borda rule.

The Borda voting rule was initially introduced by Borda [28] in 1781 and is still subject to research, see, e.g., the recent survey by Rothe [125] or the work by Neveling and Rothe [116]. In parallel, scoring protocols in general are in focus of research, too [85, 87]. Besides these scoring protocols we also study the following voting rules:

1. *Copeland$^\alpha$*: The Copeland$^\alpha$ rule is defined for every rational number $\alpha \in [0, 1]$. Given a set of candidates $C$, we run a pairwise contest for every pair of candidates $c, d \in C$ and count how many times $c$ appears in front of $d$ and how many times $d$ appears in front of $c$ in all votes from $V$. The candidate which occurs more often in front of the other one is the winner of the pairwise

contest. The contest winning candidate obtains one point, the losing candidate obtains zero points. In case of a tie, both candidates obtain $\alpha$ points. The candidate with the highest score summed over all pairwise contests wins the election.[10]

2. *Ranked pairs*: In this voting rule we step-by-step create an order over all candidates. To do so, for every two candidates $c, d \in C$ we calculate the difference between the number of votes where $c$ appears in front of $d$ and $d$ appears in front of $c$. Then, among the pairs not yet considered, we choose the pair $c, d$ with the highest difference and fix the order according to the difference, e.g., $c > d$, if this does not contradict with parts of the order established in previous steps. We continue with this approach until all pairs of candidates have been considered and we obtain a complete order over all candidates from $C$. In case of ties, we use some predefined tie-breaking mechanism. The candidate at the top of the resulting order wins the election.

3. *Plurality with runoff*: This voting rule possesses two rounds. In the first round the two candidates with the highest plurality scores proceed. For the second round, the so-called runoff, all candidates from the first round that did not proceed exit the election and we reduce all votes to the two remaining candidates. Then, the remaining candidate with the highest plurality score wins. In both rounds a predefined tie-breaking mechanism is used, if required.

4. *Veto with runoff*: Veto with runoff works analogously to plurality with runoff except that in both rounds veto instead of plurality scores are used.

5. *Simplified Bucklin voting*: For Bucklin voting we assume total preferences over $C$ and in a first step calculate the Bucklin score of every candidate in $C$. Thereby, a candidate $c$'s Bucklin score is the smallest number $b$, such that more than half of the votes in $V$ rank $c$ among the top $b$ positions. The candidates with the smallest Bucklin score win the election.

6. *Simplified fallback voting*: For fallback voting we allow partial preferences. Furthermore, every Bucklin winner is also a fallback winner, but if there is no Bucklin winner, we proceed as follows: Denote by $\ell$ the length of a longest partial preference over the candidates in $C$. Every candidate with the highest number of votes ranking her among the top $\ell$ positions is a fallback winner.

In this work, we only study simplified versions of Bucklin and fallback voting. In the original version of Bucklin voting for a candidate to win, she must, among all candidates with smallest Bucklin score $b$, also obtain the largest number of votes ranking here among the top $b$ positions, see the works by Erdélyi, Fellows, Rothe, and Schend [59, 58] for a thorough analysis of original Bucklin voting. For the original version of fallback voting, see the work by Brams and Sanver [29].

[10]The Copeland$^\alpha$ rule was introduced by Copeland [44] in 1951. However, much earlier there had already been a voting rule similar to the Copeland$^\alpha$ rule for $\alpha = 1$, today referred to as *Llull's system* [22].

With all these voting rules at hand, let us illustrate some of the just introduced rules by the subsequent example. We are given an election $(C, V)$ and determine the election's winners for various voting rules.

**Example 2.8.** Let $(C, V)$ be an election with candidates $C = \{a, b, c, d, e\}$ and 16 votes stated as follows

$$\begin{aligned} 5&: a > b > d > c > e, \\ 2&: b > c > a > e > d, \\ 3&: d > b > e > c > a, \\ 2&: a > e > b > d > c, \\ 4&: c > e > d > b > a. \end{aligned}$$

The number ahead of the colon states the number of voters participating in the election voting with the vote behind the colon. Using the plurality rule, we can see that $a$ has a plurality score of 7, $b$ has a plurality score of 2, $c$ has a plurality score of 4, $d$ has a plurality score of 3, and $e$ has a plurality score of 0, such that $a$ is the unique winner of the election. Contrarily, applying the plurality with runoff rule, in the first round the candidates $a$ and $c$ proceed into the runoff, which is then won by $c$, as $c$ achieves a score of 9 during the second round while $a$ only achieves a score of 7. If we use the Copeland$^\alpha$ rule with $\alpha = 0.5$, we obtain the subsequent scores for the pairwise contests, such that $b$ wins the election. Thereby, every row of

| | $a$ | $b$ | $c$ | $d$ | $e$ | $\sum$ |
|---|---|---|---|---|---|---|
| $a$ | - | 0 | 0 | 1 | 1 | 2 |
| $b$ | 1 | - | 1 | 1 | 1 | 4 |
| $c$ | 1 | 0 | - | 0 | 1 | 2 |
| $d$ | 0 | 0 | 1 | - | 0.5 | 1.5 |
| $e$ | 0 | 0 | 0 | 0.5 | - | 0.5 |

the previous table shows how many points the candidate obtains from the pairwise contest against the candidates in the columns.

The previous example emphasizes nicely that for the same candidates and votes one obtains different winners when applying different voting rules. This observation brings us to another fundamental question, namely, which voting rule is the "correct" one. Or, formulated differently, which of the above shown winners is the "true" winner given these votes. Of course, there is no single right answer here, as it always depends on the respective application as well as the perspective one takes. One could as well define a voting rule that represents a dictatorship and argue for this rule. To resolve this dilemma, several properties that voting rules can satisfy have been invented and defined. Among the most known properties are the *Condorcet criterion* and the *majority criterion* [22]. Thereby, an election's candidate is a *Condorcet winner* if she wins every pair-wise comparison by majority against all other candidates. Then, a voting rule satisfies the Condorcet criterion if

it guarantees to let the Condorcet winner, if one exists, win the election. A voting rule satisfies the majority criterion if it guarantees to make a candidate who is on top of more than half of all votes a winner of the election. With such properties at hand one can choose an appropriate voting rule based on the properties one wants to be satisfied. Since the later chapters of this work focus rather on computational complexity related topics with respect to elections than on topics related to the just introduced properties, we refer to the work by Baumeister and Rothe [22] for a more in-depth introduction to further properties as well as related (im)possibility results.

So far, we have only talked about elections where every voter has an equal impact on the election. Such elections, where every vote possesses the same unit-weight, are called *unweighted elections*. Besides these unweighted elections, there is also the concept of a *weighted election*. For a weighted election, every vote obtains an individual weight, indicating how much influence the corresponding voter has on the election. Formally, we can denote a weighted election by a triple $(C, V, w)$ that on top of the candidates $C$ and the votes $V$ specifies a *weight function* $w\colon V \to \mathbb{N}$ which assigns every vote $v \in V$ a nonnegative integer weight [40]. Note that every weighted election can be translated into an unweighted one by adding for every vote $v \in V$ a total of $w(v)$ unit-weight copies to the unweighted election. However, one should be aware that this translation can increase the size of the resulting, unweighted election drastically compared to the size of the weighted election, which in turn might affect computational complexities.

With the basic notations and notions defined, we now turn towards a computational complexity perspective on voting, as indicated at the beginning of this section. Similar to judgment aggregation, for a voting rule to be of practical use, its resulting winner(s) should be efficiently calculable for a given election. Bartholdi III, Tovey, and Trick [12] were the first to formulate the corresponding *winner determination* problem for a given voting rule $\mathcal{E}$.

| $\mathcal{E}$-Winner | |
|---|---|
| **Given:** | A set of candidates $C$, a list of votes $V$ over the candidates in $C$, and a distinguished candidate $c \in C$. |
| **Question:** | Is $c$ a winner of the $\mathcal{E}$-election $(C, V)$? |

For example, it is easy to see that for all scoring protocols as well as Copeland$^\alpha$ election winners can be determined efficiently. A more thorough overview of computational complexity bounds for the winner determination problem can be found in the work by Baumeister and Rothe [22]. However, computational complexity is not only used in voting to classify the winner determination problem for different voting rules. Gibbard [72] and Satterthwaite [128] showed the famous *Gibbard-Satterthwaite theorem* which, simply spoken, proved that every reasonable, preference-based voting rule is prone to *manipulation* by strategic voters [41]. Manipulation by strategic voters describes the situation when a voter is able to submit untrue preferences in order to manipulate an election's outcome towards his advantage. It was only Bartholdi III, Tovey, and Trick [11] who introduced the idea to use high compu-

tational complexities as a protection against manipulation: Even if we know from the Gibbard-Satterthwaite theorem that all of our voting rules are manipulable, we could still consider some of these rules as safe if we were to show that computing untrue preferences in order to manipulate an election's outcome is computationally infeasible [64]. Beyond manipulation, the idea of using computational complexity as a barrier was transferred to further areas such as *electoral control* [10] as well as *bribery* [63]. A notable amount of research has been and is still being done with respect to these subjects [25, 46, 58].

Nevertheless, in later chapters of our work we do neither focus on axiomatic foundations and properties of voting rules nor on manipulation, control, or bribery. Instead, we study another direction of computational complexity analysis in voting, namely *possible* and *necessary winner determination*. Given a real world election, pre-election polls, election forecasts, or similar estimates are frequently available for the election itself. Based on these incomplete data points, it is an interesting task of relevance to determine whether a distinguished candidate participating in the election can still win or will certainly loose. Konczak and Lang [98] were the first to introduce these ideas formally as decision problems to study. The question whether, given partial preferences and a set of candidates, a distinguished candidate can still win the election was formalized as the subsequent decision problem.

| $\mathcal{E}$-Possible Winner | |
|---|---|
| **Given:** | A set of candidates $C$, a set of partial votes $V$ over $C$, and a distinguished candidate $c \in C$. |
| **Question:** | Is there an extension of the partial votes in $V$ to total votes $V'$ over $C$, such that $c$ is a winner of the $\mathcal{E}$-election $(C, V')$? |

Additionally, the authors introduced the necessary winner problem which asks if, given partial preferences and a set of candidates, a distinguished candidate wins the election for every possible completion of the preferences.

| $\mathcal{E}$-Necessary Winner | |
|---|---|
| **Given:** | A set of candidates $C$, a set of partial votes $V$ over $C$, and a distinguished candidate $c \in C$. |
| **Question:** | Is $c$ a winner of every possible $\mathcal{E}$-election $(C, V')$ one can obtain by completing the partial votes from $V$ to total votes $V'$? |

While Konczak and Lang studied the complexity of their introduced decision problems for some voting rules, many more works have been published with respect to these problems, further voting rules, and extended scenarios [8, 143, 6]. For example, Betzler and Dorn [24] and Baumeister and Rothe [23] proved that the possible winner problem can be efficiently solved for the plurality and veto rule and is NP-complete for all other scoring protocols. Besides these results for the original possible and necessary winner problems, further variants of these problems have been introduced and studied. Chevaleyre, Lang, Maudet, and Monnot [39] introduced the Possible-Winner-New-Alternatives problem where the uncertainty

stems from new candidates which can be added to the election instead of from partial preferences. Baumeister, Roos, and Rothe [20] introduced the Possible-Winner-under-Uncertain-Voting-System problem where uncertainty stems from the unknown voting system instead of from partial preferences. Further variants of the possible winner problem were introduced in [18]. In Chapter 4 of our work we introduce a new variant of the possible winner problem as well, namely the Possible-Winner-with-Uncertain-Weights problem. In this problem we consider weighted elections with total preferences but unknown weights and pose the question whether there exists a weight allocation for the votes such that a distinguished candidate wins the election.

### 2.3.3 Cake-Cutting

The research area of cake-cutting is located at the intersection of computer science, economics, and mathematics. This field addresses the fair division of heterogeneous, arbitrarily divisible goods among several agents. Usually, a cake, giving the area its name, is used as a metaphor for the good to be divided. However, possible real world applications of cake-cutting range from dividing properties in case of inheritance over sharing restricted computation time on high performance computation clusters to allocating limited natural resources such as water among countries [109]. Besides the cake, the general cake-cutting scenario consists of a set of players among which the cake shall be shared and each player possesses individual preferences over the cake. The first work published in the area of cake-cutting was by Steinhaus [132] in 1948. Since then, a lot of research has been done within this field, a multitude of works have been published, and several books providing a thorough introduction to cake-cutting have been written [30, 124, 109].

Formally introducing the model of cake-cutting afterwards, we follow the notations and definitions by Lindner and Rothe [109]. The cake is represented by the unit interval $X = [0, 1]$ and this interval can be divided into arbitrary subintervals, representing individual pieces of the cake.[11] Besides the cake, we denote the players among which the cake shall be divided by $p_1, \ldots, p_n$ for $n$ players and by $\mathcal{P}$ the set of all possible pieces of cake that could be allocated. Furthermore, every player $p_i$ possesses an individual valuation function $v_i \colon \mathcal{P} \to [0, 1]$ used to describe her preferences over the cake. By default, we assume that every player only knows its own valuation function, i.e., the valuation functions are kept secret among the players. Additionally, every valuation function $v_i$ must satisfy the subsequent properties, namely

1. *normalization*, i.e., $v_i(X) = 1$ and $v_i(\emptyset) = 0$,
2. *positivity*, i.e., for every subset $A \subseteq X$ with $A \neq \emptyset$ we require $v_i(A) > 0$,
3. *additivity*, i.e., for every two disjoint subsets $A, B \subseteq X$ we require that it holds

[11] Of course, one could also use different intervals to represent the cake, e.g., $X' = [2, 4]$, but all these different possibilities can be normalized back to the unit interval.

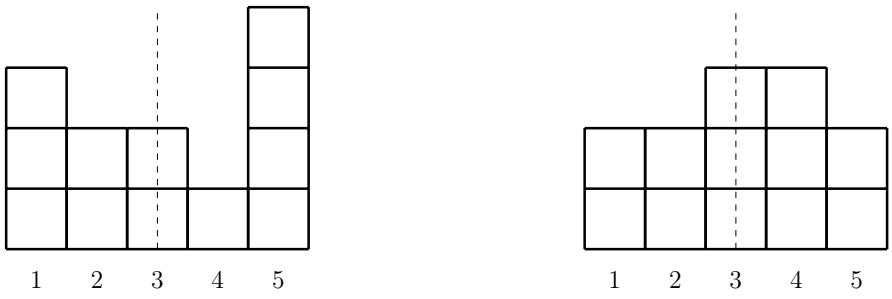
that $v_i(A \cup B) = v_i(A) + v_i(B)$, and

4. *divisibility*, i.e., for every subset $A \subseteq X$ and every $\alpha \in [0, 1]$ there is a subset $B \subseteq A$ with $v_i(B) = \alpha v_i(A)$.

As stated already at the beginning, the cake represents a heterogeneous good, i.e., a player might evaluate two pieces of the cake of same size differently. Hence, the players' evaluations do not only depend on the size of the pieces of cake but also on which part of the cake they receive. We define an *allocation* of cake $X$ among the players as a partition $X_1, \ldots, X_n$ of $X$ with $X = \bigcup_{i=1}^{n} X_i$ as well as $X_i \cap X_j = \emptyset$ for $1 \leq i < j \leq n$. Thereby, the *portion* $X_i$ is what player $p_i$ receives and it is not required that $X_i$ is a continuous piece, $X_i$ could as well consist of several pieces. Finally, a *cake-cutting protocol* describes an interactive procedure how to share a cake $X$ among the participating players. While the protocol is not aware of the players' valuation functions, it can, however, ask the players to evaluate given pieces of cake.

Having introduced all the formal definitions, let us take a look at the subsequent example that illustrates a common way how to notate valuation functions as well as a well-known cake-cutting protocol to share a cake among two players.

**Example 2.9** (Cut & choose protocol)**.** Assume that two players $p_1$ and $p_2$ want to share a cake $X = [0, 1]$. Figure 4 depicts both players' valuation functions $v_1$ and $v_2$ in the commonly used *box-based* notation. Thereby, the interval $[0, 1]$ representing




(a) $v_1$ (b) $v_2$

Figure 4: Box-based valuation functions $v_1$, $v_2$ for two players that are sharing a cake $X = [0, 1]$. The vertical, dashed line at $5/10$ represents a cut of the cake into two pieces $[0, 5/10]$ and $[5/10, 1]$.

the cake is split into five equal pieces of length $1/5$. The number of boxes per piece indicates a player's valuation of this piece. For example, player $p_1$ evaluates piece $A = [2/5, 3/5]$ by $v_1(A) = 2/12$ and $p_2$ evaluates the same piece by $v_2(A) = 3/12$. To allocate the cake among the two players, they decide to apply the *cut & choose* protocol. This protocol asks the first player to cut the cake into two pieces $X_1, X_2$ which he evaluates equally, i.e., such that $v_1(X_1) = v_1(X_2)$ holds. Afterwards, player $p_2$ can choose a piece and $p_1$ obtains the remaining piece. Following the protocol, $p_1$ cuts the cake at position $5/10$, see the dashed line in Figure 4, creating

the pieces $X_1 = [0, 5/10]$ and $X_2 = [5/10, 1]$. $p_2$ then evaluates the first piece by $v_2(X_1) = 11/24$ and the second piece by $v_2(X_2) = 13/24$, such that she chooses $X_2$ and $p_1$ obtains $X_1$ with value $v_1(X_1) = 1/2$.

The previous example introduced the box-based notation of valuation functions. This notation is a colloquial way to represent valuation functions of players. For many valuation functions this notation is sufficient. However, as we discuss in Chapter 5 of our work, this notation has its limits. In case a valuation function cannot be expressed in box-based notation, one must fall back to a more formal way of describing the function.

Furthermore, in the aforementioned example player $p_1$ was the one to cut the cake while $p_2$ was allowed to choose a piece of cake first. As the enumeration of players is arbitrary, it could as well have been the case that $p_2$ cuts the cake into two pieces and $p_1$ can choose a piece first. One can easily check that this swapped order would have resulted in two different pieces of cake being cut. Several questions follow this observation. What is a "good" allocation of a given cake to a specified set of players and how can we measure how good an existing allocation is. A lot of research has been done in the field of cake-cutting with regards to these questions and several different properties to provide answers have been suggested [123, 139, 9]. The first, most basic properties that have been defined with regards to fairness are *proportionality, super-proportionality*, and *exactness*. Given $n$ players $p_1, \ldots, p_n$ with valuation functions $v_1, \ldots, v_n$ and an allocation $X_1, \ldots, X_n$ of a cake $X$, we say that a player $p_i$'s portion is

1. *proportional*, if $v_i(X_i) \geq 1/n$,
2. *super-proportional*, if $v_i(X_i) > 1/n$, and
3. *exact*, if $v_i(X_i) = 1/n$.

The complete allocation is then said to be proportional, super-proportional, or exact, if every player's respective portion of the allocation is proportional, super-proportional, or exact. At this point we want to emphasize once more that the players' valuations are not only about the size of the respective piece they obtain but also about which part of the cake they receive. This is an important remark, as otherwise, a super-proportional allocation could not exist at all. So far, all the introduced properties are limited to a single player's valuation, e.g., a portion is proportional, if the receiving player evaluates the piece to be worth at least $1/n$ of the complete cake's value, ignoring absolutely what pieces the other players get. Consequently, there are also properties that consider all players' shares. Given again $n$ players with valuation functions $v_1, \ldots, v_n$ and an allocation $X_1, \ldots, X_n$ of a cake $X$, player $p_i$'s portion $X_i$ is

1. *envy-free*, if for every $1 \leq j \leq n$ it holds that $v_i(X_i) \geq v_i(X_j)$,
2. *super-envy-free*, if for every $1 \leq j \leq n$, $j \neq i$, it holds that $v_i(X_j) < 1/n$, and

3. *equitable* if for every $1 \leq j \leq n$ it holds that $v_i(X_i) = v_i(X_j)$.

As before, these three properties can be extended to complete allocations. The previously introduced properties all address the evaluation of an allocation from an individual player's point of view. The first characteristics only addressed the player's own piece and the later ones took also the other players' pieces into consideration. However, one can also approach the evaluation of an allocation from a holistic view. To do so, one of the most known approaches is the approach of *pareto optimality*, sometimes also called *pareto efficiency* [109]. Given $n$ players with valuation functions $v_1, \ldots, v_n$, an allocation $X_1, \ldots, X_n$ of a cake $X$ is called pareto-optimal (or pareto-efficient) if there is no other allocation $Z_1, \ldots, Z_n$ of the same cake $X$, such that

1. for all players $p_1, \ldots, p_n$ it holds that $v_i(Z_i) \geq v_i(X_i)$ and

2. there is at least one player $p_i$, $1 \leq i \leq n$, such that $v_i(Z_i) > v_i(X_i)$.

In other words, an allocation is pareto-optimal if there is no other allocation of the same cake such that no player is worse off than with the original allocation and at least one player values his alternative portion strictly better than the original one. Until now, all introduced properties have only been defined for single portions or complete allocations. However, these terms can also be extended to cake-cutting protocols. Thereby, a protocol satisfies a given property if it guarantees to always generate allocations with that property as long as all participating players follow the protocol in the designated way.

In Example 2.9 we informally introduced the cut & choose protocol. Let us now define a more formal way of specifying a cake-cutting protocol. A cake-cutting protocol consists of *rules* and *strategies*. Rules are instructions the players need to follow in order to execute the protocol and an observer can always validate that all rules were followed by the players. Contrarily, strategies are suggestions by the protocol to the players in order to ensure that the players obtain as good as possible portions of the cake, but whether a player follows a protocol's strategies or not cannot be validated by an observer who is not aware of the players' valuation functions. Referring back to the cut & choose protocol, the set of rules consists of three rules:

1. The first player must make exactly one cut to split the cake into two pieces.

2. The second player must choose one of these two pieces.

3. The first player obtains the remaining piece.

The strategies of the cut & choose protocol are as follows:

1. The first player should cut the cake in such a way into two pieces that he evaluates both pieces equally.

2. The second player should choose the piece of the cake that she prefers more.

While rules can be validated in a straightforward way, it is impossible to validate that all players stick to the strategies as their valuation functions are secret. With this background it is now clear what was meant by "designated" at the end of the previous paragraph, namely that all players follow the rules and the strategies of the protocol.

This distinction of rules and strategies raises another question: Can players obtain a better portion by deviating from a protocol's strategies? This question and related aspects address the topic of manipulability in cake-cutting. Since our chapter related to cake-cutting does not address the topic of manipulability, we do not introduce any further terms and notions related to this subject but instead refer to the work by Lindner and Rothe [109] for a more in-depth discussion.

Having introduced the general foundations as well as the desired properties for allocations of cakes, we can turn to the question how such allocations can be obtained. Many different cake-cutting protocols that more or less satisfy the previously defined properties have been introduced [30, 62, 101, 7]. But, despite knowing that a specific protocol guaranteeing to always produce an allocation with a certain set of properties exists, it is also important to know how long the protocol takes in order to end with such an allocation. So far, most of the notions and definitions we presented could also be studied from a purely mathematical or economical point of view. But with the just addressed aspects of protocol runtimes, the connection to computational complexity becomes more evident. Robertson and Webb [124] introduced a model to formalize cake-cutting protocols and based on this formalization runtimes for cake-cutting protocols can be specified. Robertson and Webb defined two types of actions that a protocol can ask from the players. The first type is the *evaluation request*, asking a player to provide the protocol with her valuation for a given piece of cake. The second type is the *cut request*, asking a player to specify a point on the cake such that cutting at this point results in a piece evaluated by the player with a specified value or to state that such a point does not exist for the asked player. Based on these request types, cake-cutting protocols can be separated into two classes. The first class contains *finite* cake-cutting protocols. Thereby, a cake-cutting protocol is finite if it guarantees to return an allocation after a finite number of requests. The second class contains *continuous* protocols and a protocol is defined as continuous if it requires the players to continuously answer requests. Continuous protocols are sometimes also called *moving-knife protocols* as one can imagine these protocols by one or multiple knifes moving across the cake awaiting the players to say "Stop" once a specified valuation criterion is satisfied, resulting in a knife cutting at this position of the cake [31, 52]. This procedure requires the players to constantly evaluate the pieces marked by the knifes' current positions.

In Chapter 5 of our work we study some axiomatic aspects related to the foundations of cake-cutting. Earlier in this section we introduced the set $\mathcal{P}$, consisting of all possible pieces of cake that a player must be able to evaluate. Intentionally, we did

not provide a formal definition for this set as several different approaches how to define this set exist in the cake-cutting literature. In the referred chapter we discuss these different approaches, their shortcomings, as well as what we suggest as an optimal definition of this set. Based on our suggestions we then analyze its impacts on other concepts from cake-cutting such as box-based valuation functions.

# 3 Complexity of Sequential Rules in Judgment Aggregation

## 3.1 Summary

In this work, we studied the family of sequential judgment aggregation rules from a computational complexity point of view. The initial motivation for this work was to determine the exact computational complexity of the Sequential Majority Winner problem which is defined as follows.

| Sequential Majority Winner ($SM$-Winner) | |
|---|---|
| **Given:** | An agenda $\Phi$, a profile $P \in \mathcal{J}(\Phi)^r$, an order $\pi$ over $\Phi_+$, and a formula $\varphi \in \Phi$. |
| **Question:** | Is $\varphi \in SM(P, \pi)$ true? |

Having studied this problem, we realized that we were able to formulate a more generic version of this decision problem by parameterizing it by its underlying judgment aggregation rule $\mathcal{K}$, yielding the $S\mathcal{K}$-Winner problem. For this general version of the decision problem we proved $\Delta_2^{\mathrm{p}}$-membership if its underlying judgment aggregation rule $\mathcal{K}$ can be computed tractably. Furthermore, we formally defined the family of sequential quota-based judgment aggregation rules, denoted by $SF_q$ for some quota rule $F_q$. Thereby, the underlying quota rules used in our work represent a special case of the general quota rules introduced by Dietrich and List [49]. Building on the $\Delta_2^{\mathrm{p}}$-membership, we proved by a polynomial-time many-one reduction from Odd Max Satisfiability that the $SF_q$-Winner problem is $\Delta_2^{\mathrm{p}}$-complete.

Having resolved the computational complexity of the winner problem, we introduced a novel, tractable counting technique that generalizes the counting technique used by Cook [42] to show that Satisfiability is NP-complete. This extended counting technique is required in the second half of our work, in which we studied possibilities of manipulative design via the processing order for sequential judgment aggregation rules. Among others, we introduced the $S\mathcal{K}$-Winner-Design problem which is defined as follows

| $S\mathcal{K}$-Winner-Design ($S\mathcal{K}$D) | |
|---|---|
| **Given:** | An agenda $\Phi$, a profile $P \in \mathcal{J}(\Phi)^r$, and a set of formulas $J \subseteq \Phi$. |
| **Question:** | Is there an order $\pi = (\varphi_1, \ldots, \varphi_m)$ over $\Phi_+$ such that $J \subseteq S\mathcal{K}(P, \pi)$? |

and proved that this problem is coNP-complete for sequential quota-based judgment aggregation rules and complete and consistent judgments $J$. Our further computational complexity results in the second half range from P-membership up to completeness in the second level of the polynomial-time hierarchy.

Finally, we explored links between the studied and other, known judgment aggregation rules, e.g., the maximal subagenda rule, see Lang and Slavkovik [104]. These

connections enabled us to transfer some of our computational complexity results, uncovering previously unknown relationships between those rules, and in some cases yielded even stricter complexities for these rules.

## 3.2 Publication

D. Baumeister, L. Boes, and R. Weishaupt. "Complexity of Sequential Rules in Judgment Aggregation". In: *Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems*. 2021, pp. 187–195

A further version of this work has been submitted to and accepted at the *8th International Workshop on Computational Social Choice* in 2021:

D. Baumeister, L. Boes, and R. Weishaupt. "Complexity of Sequential Rules in Judgment Aggregation". In: *The 8th International Workshop on Computational Social Choice (COMSOC-21)*. Ed. by B. Zwicker and R. Meir. Available online at `https://comsoc2021.net.technion.ac.il/accepted-papers/`. Haifa, Israel: Technion-Israel Institute of Technology, 2021

## 3.3 Personal Contribution

The writing of this work was conducted jointly with my co-authors Dorothea Baumeister and Linus Boes. All initial, technical results were established in equal parts by joint work with Linus Boes.

# Complexity of Sequential Rules in Judgment Aggregation

Dorothea Baumeister
Heinrich-Heine-Universität
Düsseldorf, Germany
d.baumeister@uni-duesseldorf.de

Linus Boes
Heinrich-Heine-Universität
Düsseldorf, Germany
linus.boes@uni-duesseldorf.de

Robin Weishaupt
Heinrich-Heine-Universität
Düsseldorf, Germany
robin.weishaupt@uni-duesseldorf.de

## ABSTRACT

The task in judgment aggregation is to find a collective judgment set based on the views of individual judges about a given set of propositional formulas. One way of guaranteeing consistent outcomes is the use of sequential rules. In each round, the decision on a single formula is made either because the outcome is entailed by the already obtained judgment set, or, if this is not the case, by some underlying rule, e.g. the majority rule. Such rules are especially useful for cases, where the agenda is not fixed in advance, and formulas are added one by one. This paper investigates the computational complexity of winner determination under a family of sequential rules, and the manipulative influence of the processing order on the final outcome.

## KEYWORDS

Judgment Aggregation; Computational Complexity; Winner Determination

**ACM Reference Format:**
Dorothea Baumeister, Linus Boes, and Robin Weishaupt. 2021. Complexity of Sequential Rules in Judgment Aggregation. In *Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), Online, May 3–7, 2021*, IFAAMAS, 9 pages.

## 1 INTRODUCTION

Judgment Aggregation (JA) is the task of aggregating individual judgments over logical formulas into a collective judgment set. The doctrinal paradox by Kornhauser and Sager [13] shows that if the majority rule is used, the outcome may be inconsistent, even if all underlying individual judgment sets are consistent. Since then research related to JA has been undertaken in different disciplines. The book chapter by Endriss [6] provides an overview of recent research on JA in computational social choice, where for example computer science methods are used to analyze problems originating from social choice. The investigation of JA from a computational complexity point of view has been initiated by Endriss et al. [10]. They focused on the winner problem, manipulation, and safety of the agenda problems. Subsequently, e.g. Baumeister et al. [1], Endriss and de Haan [8], and de Haan and Slavkovik [4] studied the complexity of different JA problems.

An important task is to generate consistent collective outcomes, that can, for example, be obtained through the use of sequential rules, see List [17]. A sequential rule works in rounds and uses some underlying JA rule, for example the majority rule as proposed by Dietrich and List [5] (see also Peleg and Zamir [21]). In each round the decision on one specific formula is made by checking whether either the formulas already contained in the collective outcome logically entail an assignment for the formula at hand, or otherwise, the outcome of the underlying rule for this formula will be taken. This is reasonable, since sequential procedures occur naturally by incremental decision-making. Since many real-world decisions (e.g. contract agreements) are binding, while reversing may be either favorable but expensive or impracticable, reasoning happens gradually. List [17] discusses similar use cases of such path-dependent procedures in detail. We focus on sequential rules that rely on underlying quota rules, where a formula is included in the collective outcome if a certain fraction of the judges approves it. This includes the two extreme cases where a single approval is sufficient or where an approval of all judges is needed or the common case of a majority of 2/3. Such a majority is needed for Senate votes on a presidential Impeachment, for the College of Cardinals in the papal conclave, or in some cases for constitutional amendments. Political referenda are examples of more diverse quotas.

Since JA may also be used in security applications, as mentioned by Jamroga and Slavkovik [12], it is particularly important to have consistent collective judgment sets that are efficiently computable. The complexity of winner determination for different JA rules has been studied by Endriss et al. [10] for the premise-based procedure and the distance-based procedure and by de Haan and Slavkovik [4] for scoring and distance-based rules. Along with many other rules, both, Endriss and de Haan [8] and Lang and Slavkovik [16], studied winner determination for the ranked agenda rule[1] and the maxcard subagenda rule[2], which are closely related to some of our results. In this paper we investigate the computational complexity of several problems related to winner determination for sequential JA rules that use a specific quota rule as the underlying rule. Furthermore, we study the problem of manipulative design, i.e., the question whether there is an order in which the formulas should be processed that yields some desired outcome. Additionally, we study majority-preservation for sequential JA rules, see Lang and Slavkovik [16]. The idea for sequential rules is to maintain a maximal agreement with the outcome of the majority rule (or any other underlying rule), when applied sequentially. In this context we identify a correlation between majority-preservation of sequential rules and distance based methods (in particular the maxcard subagenda rule). Our results range from membership in P to completeness in the second level of the polynomial hierarchy.

Compared to previous work on the ranked agenda rule (sequential majority rule, where the processing order is based on the majority support), see Endriss and de Haan [8] and Lang and

*Proc. of the 20th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2021), U. Endriss, A. Nowé, F. Dignum, A. Lomuscio (eds.), May 3–7, 2021, Online.*


[1] Also known in JA as Tideman's ranked pairs (see Endriss and de Haan [8]) and in similar variations as support-based procedure (see Porello and Endriss [22]) or leximax rule (see Lang et al. [15]).
[2] Also known in JA as Slater rule (see Endriss and de Haan [8]), max-num rule (see Endriss [7]) or endpoint rule (for the hamming distance as metric, see Miller et al. [18]).

Slavkovik [16], our results generalize and supplement respective complexity results, since lower bounds hold for any quota and even for a constant number of judges, implying para-NP-hardness. Additionally, we established matching upper bounds for all sequential rules that rely on a complete and complement-free rule.

## 2 PRELIMINARIES

The technical framework mainly follows the definitions in Endriss [6]. In JA we talk about a group $[r]$ of $r \in \mathbb{N}$ **judges**, where $[r]$ denotes the set $\{1, \ldots, r\}$. The judges judge over an **agenda** $\Phi$, which consists of boolean formulas in standard propositional logic. In order to avoid double negations let $\sim\varphi$ denote the complement of $\varphi$, i.e., $\sim\varphi = \neg\varphi$ if $\varphi$ is not negated, and $\sim\varphi = \psi$ if $\varphi = \neg\psi$. Thereby, we assume $\Phi$ to be finite, nonempty and closed under complement, i.e., for every $\varphi \in \Phi$ it holds that $\sim\varphi \in \Phi$. Furthermore, we assume $\Phi$ to be **nontrivial**, i.e., there exist at least two formulas $\{\varphi, \psi\} \subseteq \Phi$, such that $\{\varphi, \psi\}, \{\sim\varphi, \psi\}, \{\varphi, \sim\psi\}$ and $\{\sim\varphi, \sim\psi\}$ are consistent, and we foreclose tautologies and contradictions from $\Phi$. We split the agenda $\Phi$ into two disjoint subsets $\Phi_+$ and $\Phi_-$, where for all $\varphi \in \Phi_+$ it holds that $\sim\varphi \in \Phi_-$. Having the agenda introduced, we define an **individual judgment** $J \subseteq \Phi$ as a subset of $\Phi$. We say that $J$ is **complete**, if it holds for all $\varphi \in \Phi$ that $\varphi \in J$ or $\sim\varphi \in J$ is true. We say that $J$ is **complement-free**, if it holds for all $\varphi \in \Phi$ that $|\{\varphi, \sim\varphi\} \cap J| \leq 1$. Lastly, we define $J$ to be **consistent**, if there exists a boolean assignment for the formulas in $J$, such that all formulas are satisfied at the same time. We denote the set of all complete and consistent judgments over $\Phi$ by $\mathcal{J}(\Phi)$. For the set of judges $[r]$ we denote their **profile** of individual judgments over $\Phi$ as $P = (P_1, \ldots, P_r) \in \mathcal{J}(\Phi)^r$. We define a (resolute) **judgment aggregation rule** for an agenda $\Phi$ and $r$ judges, as a function $R\colon \mathcal{J}(\Phi)^r \to 2^\Phi$, mapping a profile $P \in \mathcal{J}(\Phi)^r$ of individual judgments to a subset $R(P)$ of $\Phi$. We say that $R$ is **complete/complement-free/consistent**, if for every profile $P \in \mathcal{J}(\Phi)^r$ it holds that $R(P)$ is complete/complement-free/consistent. Furthermore, we say that $R$ is **anonymous** if it is independent of the order of judges, i.e., $R(P) = R(P_{\pi(1)}, \ldots, P_{\pi(r)})$ for all $P \in \mathcal{J}(\Phi)^r$ permutation $\pi\colon [r] \to [r]$. Now, we define a family of JA rules. Within the subsequent definition we define a special case of the quota rules as defined by Dietrich and List [5].

*Definition 2.1 (Quota Rules).* Let $\Phi = \Phi_+ \cup \Phi_-, \Phi_+ \cap \Phi_- = \emptyset$ be an agenda, $P \in \mathcal{J}(\Phi)^r$ a profile of individual judgments and $q \in [0, 1]$. We define a **quota rule with quota** $q$ as a JA rule $F_q$ satisfying

(1) $\forall\varphi \in \Phi_+ \colon \varphi \in F_q(P) \Leftrightarrow |\{i \in [r] \mid \varphi \in P_i\}| \geq \lceil q(r+1) \rceil$ and
(2) $\forall\varphi \in \Phi_- \colon \varphi \in F_q(P) \Leftrightarrow |\{i \in [r] \mid \varphi \in P_i\}| \geq \lfloor (1-q)(r+1) \rfloor$.

Since $\lceil q(r+1) \rceil + \lfloor (1-q)(r+1) \rfloor = r+1$ holds for all $0 \leq q \leq 1$, it follows by the results from Dietrich and List [5] that all quota rules as previously defined are complete and complement-free. $\mathcal{F}$ denotes the set of all quota rules.

For an odd number of judges the majority rule equals the quota rule with quota $q = 1/2$. The difference for an even number of judges is that in case of a tie for some formula $\varphi$ the quota rule executes some tie-breaking mechanism by choosing the corresponding formula from $\Phi_-$, whereas the majority rule neglects completeness and does neither include this formula nor its negation.

We study **sequential judgment aggregation rules** in this paper. The basic idea is to ensure consistency by checking in each round whether the formulas contained in the collective outcome already fix the value for the formula at hand. This is formally denoted by the entailment relation, where $a \models b$ means that the value for $b$ is determined by $a$. To begin, we define the subsequently studied sequential JA rules in a general way.

*Definition 2.2 (Sequential $\mathcal{K}$-Judgment Aggregation Rule).* Let $\mathcal{K}$ be a complete and complement-free JA rule. Furthermore, let $\Phi$ be an agenda, $P \in \mathcal{J}(\Phi)^r$ a profile and $\pi = (\varphi_1, \ldots, \varphi_m)$ an order over $\Phi_+$. In order to obtain the aggregated judgment $S\mathcal{K}(P, \pi)$ of the **sequential $\mathcal{K}$-judgment aggregation rule**, we proceed as follows for $1 \leq i \leq m$:

(1) If either $(\varphi_1^* \wedge \ldots \wedge \varphi_{i-1}^*) \models \varphi_i$ or $(\varphi_1^* \wedge \ldots \wedge \varphi_{i-1}^*) \models \sim\varphi_i$ holds, where $\varphi_j^* \in \{\varphi_j, \sim\varphi_j\}$ is the formula added in the $j$-th iteration to $S\mathcal{K}(P, \pi)$, we add $\varphi_i$ or $\sim\varphi_i$ respectively to $S\mathcal{K}(P, \pi)$,
(2) otherwise, we add $\{\varphi_i, \sim\varphi_i\} \cap \mathcal{K}(P)$ to $S\mathcal{K}(P, \pi)$.

After $m$ iterations we obtain the final aggregated judgment $S\mathcal{K}(P, \pi)$.

As an example consider an agenda $\Phi$ with $\Phi_+ = \{a, b, a \wedge b\}$ and three judges with $J_1 = \{\neg a, b, \neg(a \wedge b)\}$, $J_2 = \{a, \neg b, \neg(a \wedge b)\}$, and $J_3 = \{a, b, a \wedge b\}$. The majority rule returns the inconsistent judgment set $\{a, b, \neg(a \wedge b)\}$. Now, consider the sequential majority rule with order $\pi = (a, a \wedge b, b)$. In the first two steps $a$ and $\neg(a \wedge b)$ are added to the outcome by majority, then the decision for $b$ is entailed by the formulas already considered and $\neg b$ is included.

Observe that by our definition (i) any output $S\mathcal{K}(P, \pi)$ is complete and consistent with respect to the agenda $\Phi$ and (ii) if $\mathcal{K}$ is anonymous then $S\mathcal{K}$ is anonymous, too. Combining (i) and (ii) with List's impossibility result [17], we obtain for underlying anonymous rules $\mathcal{K}$ that the resulting judgment of a sequential JA rule $S\mathcal{K}$ depends on the processing order over $\Phi_+$. Therefore, all previously defined (anonymous) sequential JA rules are path-dependent.

Whenever we address a sequential JA rule with respect to some JA rule $\mathcal{K}$, we assume $\mathcal{K}$ to be complement-free and complete. Subsequently, we introduce one more notation to exactly express partially aggregated judgments in order to simplify notation.

*Definition 2.3 (Partially Aggregated Judgment).* Let $\Phi$ be an agenda, $P \in \mathcal{J}(\Phi)^r$ a profile for $r$ judges, $\pi$ an order over $\Phi_+$ and $\psi \in \Phi$. We define the **partially aggregated judgment** $S\mathcal{K}^\psi(P, \pi) \subset S\mathcal{K}(P, \pi)$ as the subset of the final aggregated judgment, for which the order $\pi$ was processed until, but excluding $\psi$ or $\sim\psi$ respectively.

Observe that for every $\psi \in \Phi$ either $\psi$ itself or $\sim\psi$ appears in $\pi$, ensuring that the previous definition is well-defined. In the following, we will focus on sequential JA rules based on quota rules. For the remaining parts of the paper, we assume that the reader is familiar with the basics of computational complexity such as the classes P, NP, the polynomial hierarchy as well as polynomial-time many-one reductions $\leq_m^p$. SAT denotes the satisfiability problem and $\overline{\text{SAT}}$ its complement. For further reading, we refer to the textbook by Papadimitriou [20].

## 3 THE WINNER PROBLEM

The use of JA rules in artificial intelligence technologies raises important computational questions. As the number of judges and/or the number of formulas in the agenda may be high, it is important to design fast algorithms to determine the collective outcome.

The computational study of the winner problem for JA was initiated by Endriss et al [10]. They showed that it is polynomial-time solvable for quota rules and the premise-based procedure, while it is $\Theta_2^p$-complete for the distance-based procedure. Endriss and de Haan [8] showed that the winner problem is $\Theta_2^p$-complete for some JA rules related to known voting rules (e.g. the maxcard rule), $\Delta_2^P$-complete for the ranked agenda rule with a fixed tie-breaking and $\Sigma_2^p$-complete without a fixed tie-breaking. Lang and Slavkovik [16] defined a slightly different problem for winner determination and obtained completeness results in $\Theta_2^p$ (e.g. for the maxcard rule) and $\Pi_2^p$ (e.g. for the ranked agenda rule without tie-breaking) for majority-preserving rules. We will emphasize relationships to the former results at relevant passages. The formal definition of the winner problem for a sequential JA rule $S\mathcal{K}$ is as follows.

| $S\mathcal{K}$-Winner ($S\mathcal{K}$W) | |
|---|---|
| *Instance:* | An agenda $\Phi$, a profile $P \in \mathcal{J}(\Phi)^r$, an order $\pi$ over $\Phi_+$, and a formula $\varphi \in \Phi$. |
| *Question:* | Is $\varphi \in S\mathcal{K}(P, \pi)$ true? |

In the following, we analyze the computational complexity of this problem. We start with its upper bound.

Theorem 3.1. *$S\mathcal{K}$-Winner is in $\Delta_2^p$ if $\mathcal{K}$ is efficiently computable.*

Proof. Let $\mathcal{I} = (\Phi, P, \pi, \varphi)$ be a $S\mathcal{K}$W instance and denote the order by $\pi = (\varphi_1, \ldots, \varphi_m)$. Without loss of generality we may assume $\varphi = \varphi_j$ for one $j \in \{1, \ldots, m\}$, because if $\varphi = \sim\varphi_k$ for some $k \in \{1, \ldots, m\}$, we simply solve the instance $\mathcal{I}' = (\Phi, P, \pi, \sim\varphi)$ and invert its result.

First, we compute $\mathcal{K}(P) = \{\varphi'_1, \ldots, \varphi'_m\}$ in polynomial time. Now, for $\varphi_1$ we will use the result of $\mathcal{K}$ based on $P$ to decide whether to add $\varphi_1$ or $\sim\varphi_1$ to $S\mathcal{K}(P, \pi)$. Furthermore, denote by $\varphi^*_1, \ldots, \varphi^*_{i-1}$ the elements added to $S\mathcal{K}^{\varphi_i}(P, \pi)$ in the first $i-1$ iterations. Note, that we add any $\varphi'_i$ approved by $\mathcal{K}$, if and only if we cannot deduce $\sim\varphi'_i$ from the partially aggregated judgment. Consequently, in the $i$-th iteration, we ask whether $(\varphi^*_1 \wedge \ldots \wedge \varphi^*_{i-1}) \models \sim\varphi'_i$ holds, which is equivalent to asking whether there is no satisfying assignment for $(\varphi^*_1 \wedge \ldots \wedge \varphi^*_{i-1}) \wedge \varphi'_i$, which can be verified in coNP. Consequently, asking an NP-oracle whether this formula is satisfiable implies that $\sim\varphi'_i$ is not entailed by previously added formulas. In this case, we may add $\varphi'_i \in \mathcal{K}(P)$ directly to $S\mathcal{K}(P, \pi)$, since it is irrelevant for our purpose whether $\varphi'_i$ is deduced or added by application of $\mathcal{K}$. Therefore, we require one NP-query per iteration, except for $i = 1$. In the worst case, we have $j = m$ and must pose $m-1$ consecutive NP-queries over $m$ iterations during our computation. Note that $m-1$ is in $O(|\mathcal{I}|)$ and thus, we can solve $\mathcal{I}$ in $\Delta_2^p$. Thereby, it follows that $S\mathcal{K}\text{W} \in \Delta_2^p$ holds. □

In the construction above all queries rely on previous iterations and therefore, cannot be parallelized. Hence, $\Theta_2^p$ membership does not follow, which is in line with the general assumption of $\Theta_2^P \subset \Delta_2^p$. Now, having shown an upper bound for the computational complexity of the general winner problem, we like to introduce a lower bound for the computational complexity of the winner problem with respect to quota rules from $\mathcal{F}$. In order to do so, we first introduce the $\Delta_2^p$-complete problem Odd Max Satisfiability, as defined by Krentel [14] (see also Große et al. [11]).

| Odd Max Satisfiability (OMS) | |
|---|---|
| *Instance:* | A set $X = \{x_1, \ldots, x_n\}$ of boolean variables and a boolean formula $\alpha(x_1, \ldots, x_n)$. |
| *Question:* | Is $\alpha$ satisfiable and $x_n = 1$ in $\alpha$'s lexicographically maximum satisfying assignment $x_1 \ldots x_n \in \{0,1\}^n$? |

Theorem 3.2. *Let $F_q \in \mathcal{F}$. Then, $SF_q$-Winner is $\Delta_2^p$-complete.*

Proof. From the previous theorem we know that $SF_q\text{W} \in \Delta_2^p$ holds, since $F_q$ is efficiently computable, complement-free and complete. Therefore, it is sufficient to show OMS $\leq_{\text{m}}^{\text{p}}$ $SF_q$-Winner.

Let $\mathcal{I} = (X, \alpha)$ be an OMS instance with $X = \{x_1, \ldots, x_n\}$. We construct in time polynomial in $|\mathcal{I}|$ a $SF_q\text{W}$ instance $\mathcal{I}' = (\Phi, P, \pi, \varphi)$ as follows. Thereby, we separate the construction into two cases depending on the value of $F_q$'s quota $q$. Due to space constraints, we only present the proof for $q \leq 1/3$, the remaining case can be shown by a similar approach.

Assume $q \leq 1/3$. We define $\Phi_+ = \{\beta_1, \beta_2, \alpha', \alpha' \wedge x_1, \ldots, \alpha' \wedge x_n\}$, where $\beta_1, \beta_2$, and $\gamma$ are new variables, and $\alpha' = (\alpha \wedge \gamma) \vee \neg\beta_1 \vee \neg\beta_2$. Furthermore, we define the order $\pi$ over $\Phi_+$ as $\pi = (\beta_1, \beta_2, \alpha', \alpha' \wedge x_1, \ldots, \alpha' \wedge x_n)$ and the judges' profile $P$ as follows.

| $P$ | $\beta_1$ | $\beta_2$ | $\alpha'$ | $\alpha' \wedge x_1$ | $\ldots$ | $\alpha' \wedge x_n$ |
|---|---|---|---|---|---|---|
| $P_1$ | 0 | 1 | 1 | 1 | $\ldots$ | 1 |
| $P_2$ | 1 | 0 | 1 | 1 | $\ldots$ | 1 |

We add a formula $\psi \in \Phi_+$ to the aggregated judgment $F_q(P)$ if and only if $|\{i \in [r] \mid \psi \in P_i\}| \geq \lceil q(r+1) \rceil$ holds. For $r = 2$ and $q \leq 1/3$ we have $\lceil q(r+1) \rceil \leq 1$, so that $F_q(P) = \Phi_+$ holds.

We set $\varphi = \alpha' \wedge x_n$. Furthermore, no consistency condition is violated since $\alpha'$ can be satisfied for every individual judgment via $\beta_1, \beta_2$, even when $\alpha$ is unsatisfiable. In order to prevent $\alpha'$ from turning into a tautology when $\alpha$ is one, we added $\gamma$.

Now, we prove that $\mathcal{I} \in \text{OMS} \Leftrightarrow \mathcal{I}' \in SF_q$-Winner holds. For the direction from left to right assume that $\mathcal{I}$ is a yes-instance. After the first two iterations of the $SF_q$-rule we have $SF_q^{\alpha'}(P, \pi) = \{\beta_1, \beta_2\}$. By assumption, there exists a satisfying assignment for $\alpha$ and trivially also for $\neg\gamma$. Therefore, in the third round we can neither entail $\neg\alpha' \in SF_q(P, \pi)$ nor $\alpha' \in SF_q(P, \pi)$. Thus, we add $\alpha'$ by applying the $F_q$-rule. Consequently, after the third iteration we have $SF_q^{\alpha' \wedge x_1}(P, \pi) = \{\beta_1, \beta_2, \alpha'\}$. From this fact it follows that $SF_q^{\alpha' \wedge x_1}(P, \pi) \models \alpha \wedge \gamma \models \alpha, \gamma$ holds, which is in accordance with our assumption that $\alpha$ is satisfiable. Now, we would like to decide whether to add $\alpha' \wedge x_1$ or $\neg(\alpha' \wedge x_1)$ to $SF_q(P, \pi)$. Given the current aggregated judgment and knowing that $\gamma \equiv$ true, it holds that $\alpha' \wedge x_1 = [(\alpha \wedge \gamma) \vee \neg\beta_1 \vee \neg\beta_2] \wedge x_1 \equiv \alpha \wedge x_1$. Furthermore, knowing from $\alpha \wedge \gamma \equiv \alpha' \in SF_q(P, \pi)$ that $\alpha$ should be true, we distinguish three cases for $\alpha \wedge x_1$: (i) If $x_1 = 1$ is the only option for a satisfying assignment of $\alpha$, we can deduce $\alpha' \wedge x_1 \in SF_q(P, \pi)$. (ii) If $x_1 = 0$ is the only option for a satisfying assignment of $\alpha$, we can deduce $\neg(\alpha' \wedge x_1) \in SF_q(P, \pi)$. (iii) If there are satisfying assignments for $\alpha$ with both, $x_1 = 1$ and $x_1 = 0$, we must apply the $F_q$-rule and obtain $\alpha' \wedge x_1 \in SF_q(P, \pi)$. Note that the last option always favors the bigger satisfying assignment, i.e., preferring $x_1 = 1$ over $x_1 = 0$. We can apply the previous argument for $j \in \{1, \ldots, n\}$ and deduce for all formulas $\alpha' \wedge x_j$ whether to add them or their

corresponding negation $\neg(\alpha' \wedge x_j)$ to $SF_q(P,\pi)$. Doing so yields a maximum satisfying assignment for $\alpha$, represented by $[x_i = 1] \Leftrightarrow [\alpha' \wedge x_i \in SF_q(P,\pi)]$. By assumption, we know that $x_n = 1$ holds for a maximum satisfying assignment of $\alpha$. Thus, $\alpha' \wedge x_n \in SF_q(P,\pi)$ holds after the last iteration and therefore, $\mathcal{I}' \in SF_q$-Winner is true.

For the direction from right to left assume now that $\mathcal{I}$ is a no-instance. We study two separate cases.

**Case 1:** $\alpha$ is satisfiable but for its maximum satisfying assignment $x_n = 0$ holds. In the third iteration we add $\alpha'$ to $SF_q(P,\pi)$. As already argued in the first part of the proof, for $1 \leq j \leq n$ we add $\alpha' \wedge x_j$ to $SF_q(P,\pi)$ if and only if $x_j = 1$ holds in $\alpha$'s maximum satisfying assignment. By assumption, we know that $x_n = 0$ is true in $\alpha$'s maximum satisfying assignment. Therefore, we end up with $\alpha' \wedge x_n \notin SF_q(P,\pi)$ and can conclude that $\mathcal{I}' \notin SF_q$-Winner holds.

**Case 2:** $\alpha$ is not satisfiable. After the first two iterations of the $SF_q$-rule we have $SF_q^{\alpha'}(P,\pi) = \{\beta_1, \beta_2\}$. By assumption, in the third iteration it holds that

$$\alpha' = (\alpha \wedge \gamma) \vee \neg\beta_1 \vee \neg\beta_2 \equiv (\text{false} \wedge \gamma) \vee \neg\beta_1 \vee \neg\beta_2 \equiv \text{false}.$$

Consequently, we deduce that $\neg\alpha'$ must hold and thus add $\neg\alpha'$ to $SF_q(P,\pi)$. Obviously, this leads to the fact that we add $\neg(\alpha' \wedge x_j)$ to $SF_q(P,\pi)$ for $1 \leq j \leq n$. Therefore, we have $\alpha' \wedge x_n \notin SF_q(P,\pi)$ and hence, $\mathcal{I}' \notin SF_q$-Winner.

Finally, we have $\mathcal{I} \in$ OMS if and only if $\mathcal{I}' \in SF_q$-Winner and obtain OMS $\leq_m^p SF_q$-Winner. □

Endriss and de Haan [8] showed that the winner problem for the ranked agenda rule (with fixed tie-breaking) is $\Delta_2^p$-hard. However, the corresponding proof requires a linear number of judges. We note that slightly modifying our previous proof by adding a third judge, supporting both, $\beta_1$ and $\beta_2$, but no other formula, allows us to reuse the same proof (i.e., the given order $\pi$) for the ranked agenda rule. This yields an even stricter result for the ranked agenda's winner problem's complexity, namely para-$\Delta_2^p$-hardness with respect to the number of judges.

Corollary 3.3. *The winner problem for the ranked agenda rule with fixed tie-breaking is para-$\Delta_2^p$-hard when parameterized by the number of judges.*

Note that our lower bound proofs in Section 5 may be adapted in a similar way (by adding a third judge only approving corresponding $\beta_j$) to also handle the ranked agenda rule.

## 4 COUNTING TECHNIQUE

Within this section, we introduce a polynomial-time computable technique used to construct a boolean formula $\psi_k^B$. The formula is able to count the number of satisfied boolean variables for a given boolean assignment $T$ of a set of boolean variables $B$ in the sense that a truth assignment evaluates the formula to true if and only if at most $k \in \mathbb{N}$ of the variables in $B$ for $T$ are true.

In some sense our technique generalizes the already known technique used by Cook in his famous theorem to prove that SAT is NP-complete, cf. [2]. Cook's technique describes an approach how to formulate a boolean formula for a set of boolean variables which is true if and only if exactly one of the boolean variables is true.

Lemma 4.1. *Let $B = \{x_1, \ldots, x_n\}$ be a set of boolean variables and $k \leq n$. We can construct a formula $\psi_k^B$ from a set of boolean variables $B'$ with $|B'| = nk$ in time polynomial in n, such that $\psi_k^B$ evaluates to true if and only if at most k of the n boolean variables in B are set to true.*

Proof. In a first step, we create $k$ copies $\{x_i^1, \ldots, x_i^k\}$ for every boolean variable $x_i$ in $B$. Then, we define a boolean formula $X_i$ for every $1 \leq i \leq n$ as follows $X_i = \left[\bigvee_{j\in[k]} \left(x_i^j \wedge \bigwedge_{\ell\in[k]\setminus\{j\}} \neg x_i^\ell\right)\right] \vee \left[\bigwedge_{j\in[k]} \neg x_i^j\right]$. Consequently, $X_i$ is satisfied if and only if at most one of the $k$ copies of $x_i$ is satisfied. Note that every $X_i$ can be constructed in time in $O(n^2)$ since $|X_i| = k(k+1) \leq n(n+1)$ holds.

In a second step, we construct $k$ boolean formulas $Y_j$ for $1 \leq j \leq k$ as follows $Y_j = \left[\bigvee_{i\in[n]} \left(x_i^j \wedge \bigwedge_{\ell\in[n]\setminus\{i\}} \neg x_\ell^j\right)\right] \vee \left[\bigwedge_{i\in[n]} \neg x_i^j\right]$. Thereby, $Y_j$ is satisfied if and only if at most one of the $n$ variables in the $j$-th set of copies $\{x_1^j, \ldots, x_n^j\}$ is satisfied. Note that we can also construct $Y_j$ in time in $O(n^2)$ since $|Y_j| = n(n+1)$ holds.

In a third step, we define two more boolean formulas, namely $Y = \bigwedge_{j=1}^k Y_j$ and $X = \bigwedge_{i=1}^n X_i$. Consequently, $Y$ is satisfied if and only if for every $j$, $1 \leq j \leq k$, at most one variable in the set $\{x_1^j, \ldots, x_n^j\}$ is satisfied. Analogously, $X$ is satisfied if and only if at most one of the copies for every $x_i$, $1 \leq i \leq n$, is satisfied. Finally, setting $\psi_k^B = Y \wedge X$ obviously completes the construction.

It remains to show the correctness of the construction. To do so, first we explain how to derive a boolean assignment $T'$ for $B' = \{x_1^1, \ldots, x_1^k, \ldots, x_n^1, \ldots, x_n^k\}$ out of a boolean assignment $T$ for $B = \{x_1, \ldots, x_n\}$. Therefore, denote by $\rho(B,T) = \{x \in B \mid T(x) = \text{true}\}$ the set of variables set to true by $T$. We construct $T'$ as follows. Write $\rho(B,T) = \{x_{i_1}, \ldots, x_{i_m}\}$ for $m \leq n$. For $1 \leq j \leq m$, we set $x_{i_j}^{(j \bmod k)+1}$ to true and all other variables in $B'$ to false.

The formal proof of correctness is omitted due to space constraints. □

We will use this technique as follows. Let $B = \{x_1, \ldots, x_n\}$ be a set of boolean variables, $k \in \mathbb{N}$ and $\alpha(B)$ some boolean formula over $B$. At some point, we must know whether a given assignment $T$ satisfies $\alpha(B)$, while no more than $k$ of the boolean variables in $B$ should be set to true. In order to decide this fact efficiently, we first globally replace each variable $x_i \in B$ that appears in $\alpha$ by $\bigvee_{j\in[k]} x_i^j$ and denote the result as $\alpha_k$. Then, we construct a new boolean formula $\alpha_k' = \alpha_k \wedge \psi_k^B$ and check whether $\alpha_k'$ is true for the corresponding assignment $T'$. If this is the case, we know that $\alpha(T(B))$ is true, while no more than $k$ of the $n$ variables in $B$ are true for $T$. In order to keep our notation as simple as possible, we write $\alpha' = \alpha \wedge \psi_k^B$.

## 5 PROBLEMS OF MANIPULATIVE DESIGN

While the usage of sequential rules guarantees consistency, at the same time the gradual aggregation approach leads to problems of manipulative design for anonymous underlying rules. Following the impossibility result by List [17], sequential quota rules are path-dependent, i.e., the aggregated judgment is determined by the processing order of formulas and might be altered at will if said order is chosen accordingly. Realizing the amount of power

a manipulator in control over the processing order has, we study how hard it is to compute whether at least one (respectively every) order guarantees a partial judgment to be included into the aggregated one. Although List already proposed said approach as *Manipulation by Agenda Setting*, we deviate in studying two variants. In particular, we study the $S\mathcal{K}$-Winner-Design and the $S\mathcal{K}$-Winner-Robustness problem and will show that it is more inefficient for sequential quota rules to solve proposed problems of manipulative design than the corresponding winner problem. The formal definition of the Winner-Design problem is as follows for a given sequential JA rule $S\mathcal{K}$.

| $S\mathcal{K}$-Winner-Design ($S\mathcal{K}$D) | |
|---|---|
| *Instance:* | An agenda $\Phi$, a profile $P \in \mathcal{J}(\Phi)^r$, and a set of formulas $J \subseteq \Phi$. |
| *Question:* | Is there an order $\pi = (\varphi_1, \ldots, \varphi_m)$ over $\Phi_+$ such that $J \subseteq S\mathcal{K}(P, \pi)$? |

Analogously we formulate the almost complementary decision problem S$\mathcal{K}$-Winner-Robustness (S$\mathcal{K}$R). The input remains unchanged but the question is whether $J \subseteq S\mathcal{K}(P, \pi)$ holds for every processing order $\pi$ over $\Phi_+$. In order to determine the computational complexity of S$\mathcal{K}$D and S$\mathcal{K}$R, we require some notation.

*Definition 5.1.* Let $\mathcal{K}$ be a complete and complement-free JA rule, $\Phi$ an agenda, and $P \in \mathcal{J}(\Phi)^r$ a profile for $r$ judges. Furthermore, slightly abusing notation, let $\pi = (\varphi_1, \ldots, \varphi_m)$ be an order over $\mathcal{K}(P)$ and denote by $S\mathcal{K}(P, \pi)$ the corresponding aggregated judgment. Let $K_\pi = \mathcal{K}(P) \cap S\mathcal{K}(P, \pi)$ denote the set of formulas in the aggregated judgment also supported by $\mathcal{K}$, and $D_\pi = S\mathcal{K}(P, \pi) \setminus \mathcal{K}(P)$ those not supported by $\mathcal{K}$. For $K_\pi = \{k_1, \ldots, k_p\}$ and $D_\pi = \{d_1, \ldots, d_{m-p}\}$ let $(K_\pi, D_\pi) = (k_1, \ldots, k_p, d_1, \ldots, d_{m-p})$ denote an order, where all formulas in $K_\pi$ are permuted arbitrarily at the first $p$ places.

This enables us to formulate the following lemma.

Lemma 5.2. *Let $\mathcal{K}$ be a complete and complement-free JA rule, $\Phi$ an agenda and $P \in \mathcal{J}(\Phi)^r$ a profile for $r$ judges. Then, for every order of the form $\pi' = (K_\pi, D_\pi)$ it holds that $S\mathcal{K}(P, \pi') = S\mathcal{K}(P, \pi)$.*

The intuition is, that we can rearrange every order $\pi$ in such a way that all formulas supported by $\mathcal{K}$ are at the beginning of $\pi$ and all remaining formulas follow afterwards. Hence, instead of looking for a specific order it is sufficient to search for a consistent subset $K \subseteq \mathcal{K}(P)$, such that $K \models \bigwedge_{\varphi \in J} \varphi$ holds. Doing so enables us to solve a $S\mathcal{K}$-Winner-Design instance by setting $\pi = (K, J, \ldots)$.

Note that for $q = 1/2$, the problems $SF_q$-Winner-Design and $SF_q$-Winner-Robustness are closely related to the winner determination problem for the ranked agenda rule without fixed tie-breaking as studied by Endriss and de Haan [8] and Lang and Slavkovik [16]. Both investigate hardness for similar decision problems, where the processing order is additionally required to be in accordance with the number of supporting judges (i.e., for any order $\pi = (\varphi_1, \ldots, \varphi_m)$ over $F_{1/2}(P)$ it holds that $|\{i \in [r] \mid \varphi_j \in P_i\}| \geq |\{i \in [r] \mid \varphi_{j+1} \in P_i\}|$). We continue to study the complexity for two widely separated cases, namely manipulative design for complete judgment sets (Section 5.1) and for single formulas (Section 5.2). An overview of our results is given in Table 1.

## 5.1 Manipulative Design for Judgment Sets

First, let us investigate the introduced problems of manipulative design for a given judgment which is complete and consistent. Note that we do not consider inconsistent judgments, since those are neither desirable nor a possible output. The ensuing theorem derives an upper bound of coNP for a broad class of sequential JA rules.

Theorem 5.3. *For every polynomial-time computable JA rule $\mathcal{K}$ that is complete and complement-free, it holds that $S\mathcal{K}$D $\in$ coNP if the desired subset of formulas equals a complete and consistent judgment $J \in \mathcal{J}(\Phi)$.*

Proof. We precompute $K = J \cap \mathcal{K}(P)$ and $D = J \setminus \mathcal{K}(P)$ in polynomial time. Since $J \in \mathcal{J}(\Phi)$, $K$ and $D$ are consistent. Following Lemma 5.2 it is sufficient to verify whether each formula in $D$ can be derived from $K$, since we then may construct an order of the form $\pi' = (K, D)$. Hence, we have to check whether $\left(\bigwedge_{\varphi \in K} \varphi\right) \models \left(\bigwedge_{\psi \in D} \psi\right)$. This is equivalent to checking whether there is no assignment satisfying $\left(\bigwedge_{\varphi \in K} \varphi\right) \wedge \neg \left(\bigwedge_{\psi \in D} \psi\right)$ and hence in coNP. □

For the class of quota rules the following theorem establishes the matching lower bound and proves coNP-hardness.

Theorem 5.4. *For every quota rule $F_q \in \mathcal{F}$ and every given complete and consistent judgment $J \in \mathcal{J}(\Phi)$ it is coNP-complete to solve the corresponding $SF_q$D problem.*

Proof. Recall that we assume every quota rule $F_q$ to be complete and complement-free for every quota $q$. To show coNP-hardness, we reduce a $\overline{\text{SAT}}$ instance $\mathcal{I} = (\alpha)$ to a $SF_q$D instance $\mathcal{I}' = (\Phi, P, J)$. We define $\Phi_q = \{(\alpha \wedge \gamma) \vee \neg\beta_1 \vee \neg\beta_2, \beta_1, \beta_2\}$, where $\gamma$, $\beta_1$, and $\beta_2$ are new literals, and choose $\Phi_+ = \Phi_q$ for $q \leq 1/3$ and $\Phi_- = \Phi_q$ otherwise. We consider a profile consisting of two judges with $P_i = \{(\alpha \wedge \gamma) \vee \neg\beta_1 \vee \neg\beta_2, \beta_i, \neg\beta_{3-i}\}$ for $i \in [2]$. Note that by construction it holds that $F_q(P) = \Phi_q$. Lastly, we set $J = P_1$ and show that equivalence holds. For the direction from left to right assume $\mathcal{I}$ is a yes-instance and thus, $\alpha$ is unsatisfiable. Choosing the order $\pi = ((\alpha \wedge \gamma) \vee \neg\beta_1 \vee \neg\beta_2, \beta_1, \beta_2)$ over $\Phi_q$ results in $SF_q(P, \pi) = J$. For the direction from right to left assume $\mathcal{I}$ is a no-instance and thus, $\alpha$ is satisfiable. Then, $F_q(P)$ is already consistent and $SF_q(P, \pi) = F_q(P) \neq J$ holds for every order $\pi$. Together with Theorem 5.3 we obtain coNP-completeness. □

Turning to the robustness problem, we require that the desired judgment set $J$ is contained in the collective outcome for every possible order. This is only possible if each of the formulas is contained in the collective judgment set of the underlying formula.

Theorem 5.5. *For every agenda $\Phi$, profile $P \in \mathcal{J}(\Phi)^r$ and complete and consistent judgment $J \in \mathcal{J}(\Phi)$, the corresponding $S\mathcal{K}$R-instance $(\Phi, P, J)$ is satisfiable if and only if $\mathcal{K}(P) = J$ for a complete and complement-free procedure $\mathcal{K}$.*

Note that for efficiently computable underlying rules and particularly for sequential quota rules $SF_q$ the corresponding problem is decidable in P.

**Table 1: Summary of complexity results for different problems regarding sequential JA rules $SF_q$.**

| Winner | Winner-Design | | Winner-Robustness | | Supported-Judgment |
|---|---|---|---|---|---|
| | $J \in \mathcal{J}(\Phi)$ | $\varphi \in \Phi$ | $J \in \mathcal{J}(\Phi)$ | $\varphi \in \Phi$ | |
| $\Delta_2^p$-c., Thm. 3.1, 3.2 | coNP-c., Thm. 5.3, 5.4 | $\Sigma_2^p$-c., Thm. 5.7, 5.9 | in P, Thm. 5.5 | $\Pi_2^p$-c., Lem. 5.6 | NP-c., Thm. 5.12, 5.13 |

## 5.2 Manipulative Design for Single Formulas

Before investigating the complexity of S$\mathcal{K}$D and S$\mathcal{K}$R separately, we want to point out that they are tied closely together, when testing whether a single formula is in the aggregated judgment.

Lemma 5.6. *For every complete and complement-free procedure $\mathcal{K}$, every agenda $\Phi$, every profile $P \in \mathcal{J}(\Phi)^r$ and every formula $\varphi \in \Phi$, it holds that $(\Phi, P, \{\varphi\}) \in S\mathcal{K}R \Leftrightarrow (\Phi, P, \{\sim\varphi\}) \in \overline{S\mathcal{K}D}$.*

Above lemma follows from complement-freeness and completeness and has also been shown by Lang and Slavkovik [16]. In the following, we will only show complexity results for S$\mathcal{K}$D, while results for S$\mathcal{K}$R follow directly. We continue to establish upper bounds.

Theorem 5.7. *For every polynomial-time computable, complete and complement-free JA rule $\mathcal{K}$ and a judgment $J = \{\varphi\} \subset \Phi$ containing a single formula, it holds that $S\mathcal{K}$-Winner-Design $\in \Sigma_2^p$.*

Proof. In order to solve an instance $\mathcal{I} = (\Phi, P, \{\varphi\})$ of the decision problem $S\mathcal{K}$-Winner-Design, we must determine whether there exists an order $\pi$ such that $\varphi \in S\mathcal{K}(P, \pi)$ holds. Exploiting our previous observations, we know from Lemma 5.2 that it is sufficient to identify a consistent subset $K \subseteq \mathcal{K}(P)$ with $K \models \varphi$.

Thus, we first calculate $\mathcal{K}(P)$ in polynomial time and can nondeterministically guess a subset $K = \{\varphi_1, \ldots, \varphi_k\} \subseteq \mathcal{K}(P)$. Next, we verify whether $K$ is consistent by asking our NP-oracle whether there exists a satisfying assignment for $\varphi_1 \wedge \ldots \wedge \varphi_k$. In a last step, we must determine whether $K \models \varphi$ holds. Thereby, we have $(\varphi_1 \wedge \ldots \wedge \varphi_k) \models \varphi$. To determine whether this formula is satisfiable can again be solved in coNP. Consequently, we can pose a second NP-query to find out whether $K$ entails $\varphi$, resulting in $\varphi \in S\mathcal{K}(P, \pi)$ for $\pi = (K, \varphi, \ldots)$. Overall, we require a polynomial amount of nondeterministic computation steps as well as two NP-oracle queries to calculate an answer for $\mathcal{I}$ and thus, $S\mathcal{K}$-Winner-Design $\in \Sigma_2^p$ holds. □

Combining the former theorem with Lemma 5.6, we derive the following corollary.

Corollary 5.8. *For every complete and complement-free JA rule $\mathcal{K}$ computable in polynomial time and a judgment $J = \{\varphi\} \subset \Phi$, it holds that $S\mathcal{K}$-Winner-Robustness $\in \Pi_2^p$.*

In order to identify lower bounds for sequential quota rules, let us first define the decision problem Succinct Set Cover (SSC), which was proven to be $\Sigma_2^p$-complete by Umans [23]. The instance consists of a collection of 3-DNF formulas $S = \{\varphi_1, \ldots, \varphi_n\}$ over $m$ variables and $k \in \mathbb{N}$. The question is whether there is a subset $N' \subseteq [n]$ with $|N'| \leq k$ and $\bigvee_{i \in N'} \varphi_i \equiv$ true?

Theorem 5.9. *For every quota rule $F_q \in \mathcal{F}$ and a judgment $J = \{\varphi\} \subset \Phi$ consisting of a single formula, it holds that the problem $SF_q$-Winner-Design is $\Sigma_2^p$-complete.*

Proof. Due to Theorem 5.7 it is enough to show $\Sigma_2^p$-hardness. We reduce Succinct Set Cover to $SF_q$-Winner-Design. Let $\mathcal{I} = (\{\varphi_1, \ldots, \varphi_n\}, k)$ be a SSC instance. To construct $\mathcal{I}' = (\Phi, P, \{\varphi\})$, we first introduce some auxiliary variables. Let $B = \{x_1, \ldots, x_n\}$ be a set of boolean literals, $\psi_k^B$ defined as described in Section 4 and $\varphi_i' = (\varphi_i \wedge x_i)$ for $1 \leq i \leq n$. For our construction we set $\varphi = \psi_k^B \wedge \left[\left(\bigvee_{i \in [n]} \varphi_i'\right) \vee \gamma\right] \wedge \beta_1 \wedge \beta_2$ and $\Phi_q = B \cup \{\beta_1, \beta_2\} \cup \{\psi_k^B \vee \neg\beta_1 \vee \neg\beta_2, \sim\varphi\}$ with new literals $\beta_j$ and $\gamma$. Note that by including $\gamma$, the agenda cannot contain any contradictions or tautologies. More precisely, both $\psi_k^B, \varphi$ and their negations are satisfiable, even if every $\varphi_i$ is a contradiction. The judges' profile consists of two judgments $P_i = \Phi_q \setminus \{\beta_i\} \cup \{\neg\beta_i\}$ for $i \in \{1, 2\}$ and the individual judgments' consistency is not violated, since $\sim\varphi$ is always satisfiable by any $\neg\beta_j$. Finally, we set $\Phi_+ = \Phi_q$ for $q \leq 1/3$ and $\Phi_- = \Phi_q$ otherwise. By construction it holds that $F_q(P) = \Phi_q$ and, slightly abusing notation, we consider any order $\pi$ over $\Phi_q$ instead of $\Phi_+$. Clearly, this construction can be done in polynomial time. Subsequently, we prove $\mathcal{I} \in$ SSC $\Leftrightarrow \mathcal{I}' \in \mathrm{SF}_q$D.

($\Rightarrow$) Assume $\mathcal{I}$ is a yes-instance. Consequently, there exists a set $N' = \{i_1, \ldots, i_m\} \subseteq [n]$ with $m \leq k$ such that $\bigvee_{i \in N'} \varphi_i \equiv$ true. As order we choose $\pi = (\beta_1, \beta_2, \psi_k^B \vee \neg\beta_1 \vee \neg\beta_2, x_{i_1}, \ldots, x_{i_m}, \sim\varphi, \ldots)$, where the order of the elements after $\sim\varphi$ is irrelevant. Applying the $SF_q$-rule, we may add each formula in the first $m + 3$ iterations by using the quota rule $F_q$, since $\psi_k^B$ and $m \leq k$ variables from $B$ are satisfiable simultaneously, even if both $\beta_j$ are set to true. Now, we show that $\varphi = \psi_k^B \wedge \left[\left(\bigvee_{i \in [n]} \varphi_i'\right) \vee \gamma\right] \wedge \beta_1 \wedge \beta_2$ may be deduced from the initial assumption by showing that each formula in $\{\psi_k^B, \bigvee_{i \in [n]} \varphi_i', \beta_1, \beta_2\}$ can be deduced separately. First, note that each $\beta_j$ trivially entails itself and $\beta_1 \wedge \beta_2 \wedge \left(\psi_k^B \vee \neg\beta_1 \vee \neg\beta_2\right) \models \psi_k^B$ holds. For the remaining formula it holds that

$$
\begin{aligned}
&\bigwedge_{i \in N'} x_i \Rightarrow \bigvee_{i \in [n]} \varphi_i' \\
\Leftrightarrow &\bigvee_{i \in N'} \neg x_i \vee \bigvee_{i \in [n]} (\varphi_i \wedge x_i) \\
\Leftrightarrow &\bigvee_{i \in N'} ((\neg x_i \wedge \varphi_i) \vee (\neg x_i \wedge \sim\varphi_i)) \vee \bigvee_{i \in [n]} (\varphi_i \wedge x_i) \\
\Leftrightarrow &\bigvee_{i \in N'} \varphi_i \vee \bigvee_{i \in N'} (\neg x_i \wedge \sim\varphi_i) \vee \bigvee_{i \in [n] \setminus N'} (\varphi_i \wedge x_i), \qquad (1)
\end{aligned}
$$

where the left disjunction in (1) already is a tautology by assumption. Consequently, it holds that $SF_q^{\sim\varphi}(P, \pi) \Rightarrow \varphi$. Hence, we conclude $\varphi \in SF_q(P, \pi)$, resulting in $\mathcal{I}' \in SF_q$D.

($\Leftarrow$) Assume $\mathcal{I}$ is a NO-instance. Consequently, there does not exist any $N' \subseteq [n]$ with $|N'| \leq k$, such that $\bigvee_{i \in N'} \varphi_i \equiv \text{TRUE}$ holds. By contradiction, we assume $\mathcal{I}'$ to still be a YES-instance. Then, there exists an order $\pi$ over $\Phi_q$ such that

$$\varphi = \psi_k^B \wedge \left[\left(\bigvee_{i \in [n]} \varphi_i'\right) \vee \gamma\right] \wedge \beta_1 \wedge \beta_2 \in SF_q(P, \pi)$$

holds. We deduce that $\beta_1, \beta_2 \in SF_q(P, \pi)$ and $\psi_k^B \vee \neg\beta_1 \vee \neg\beta_2 \in SF_q(P, \pi)$ hold as well due to consistency. Hence, at most $k$ of the variables $x_i$, $1 \leq i \leq n$, are satisfied. Let us denote the satisfied variables by $M = \{x_{i_1}, \ldots, x_{i_{k'}}\}$ and the unsatisfied variables by $B \backslash M = \{x_{i_{k'+1}}, \ldots, x_{i_n}\}$. Furthermore, we can imply the following out of $\varphi \in SF_q(P, \pi)$:

$$\begin{aligned}
\text{TRUE} &\equiv \left[\bigvee_{i \in [n]} \varphi_i'\right] \vee \gamma \equiv \left[\bigvee_{i \in [n]} (\varphi_i \wedge x_i)\right] \vee \gamma \\
&\equiv \left[\bigvee_{i \in M} (\varphi_i \wedge x_i)\right] \vee \left[\bigvee_{i \in [n] \backslash M} (\varphi_i \wedge x_i)\right] \vee \gamma \\
&\equiv \left[\bigvee_{i \in M} (\varphi_i \wedge \text{TRUE})\right] \vee \left[\bigvee_{i \in [n] \backslash M} (\varphi_i \wedge \text{FALSE})\right] \vee \gamma \\
&\equiv \left[\bigvee_{i \in M} \varphi_i\right] \vee \text{FALSE} \vee \gamma \equiv \left[\bigvee_{i \in M} \varphi_i\right] \vee \gamma.
\end{aligned}$$

Yet, we know that $\varphi$ must have been entailed by previously added formulas because $\varphi \notin F_q(P)$. Hence, we conclude that for the given order $\pi$ it holds that $SF_q^{\sim\varphi}(P, \pi) \models (\bigvee_{i \in M} \varphi_i) \vee \gamma$, although neither $\gamma$ nor any $\varphi_i$ shares any literals with formulas from $SF_q^{\sim\varphi}(P, \pi)$. Overall, $(\bigvee_{i \in M} \varphi_i) \vee \gamma$ can only be entailed if the disjunction contains a tautology. Since $\gamma$ is a literal, this implies that $\bigvee_{i \in M} \varphi_i \equiv \text{TRUE}$ with $|M| \leq k$ would be a solution to $\mathcal{I}$, which is a contradiction to our assumption. Therefore, such an order $\pi$ cannot exist and $\mathcal{I}'$ must be a NO-instance, too. □

Again, we derive a corollary for $SF_q$R from the previous theorem and Lemma 5.6.

COROLLARY 5.10. *For every quota rule $F_q \in \mathcal{F}$ and a judgment $J = \{\varphi\} \subset \Phi$, it holds that $SF_q$-WINNER-ROBUSTNESS is $\Pi_2^p$-complete.*

Endriss and de Haan [8] investigate the complexity of existential winner-determination for the ranked agenda rule without a fixed tie-breaking which is shown to be $\Sigma_2^p$-hard. Similarly to corollary 3.3, we may improve this result, as our proof of Theorem 5.9 can easily be adapted (by adding a third judge only approving $\beta_j$) to also hold for the ranked agenda rule without fixed tie-breaking.

COROLLARY 5.11. *The winner problem for the ranked agenda rule without fixed tie-breaking is para-$\Sigma_2^p$-hard when parameterized by the number of judges.*

## 5.3 Supported Judgment

We conclude this section by formulating a problem, which formally relates to problems of manipulative design, although it is clearly motivated contrarily. In terms of acceptance, it is desirable for an aggregated judgment to be reasonable for the participating judges. Hence, for sequential JA rules it should be preferable to choose an order such that at least $k$ formulas supported by a rule $\mathcal{K}$ are included in the aggregated judgment.

| $S\mathcal{K}$-SUPPORTED-JUDGMENT ($S\mathcal{K}$SJ) | |
|---|---|
| *Instance:* | An agenda $\Phi$ with $\lvert\Phi_+\rvert = m$, a profile $P \in \mathcal{J}(\Phi)^r$ for $r$ judges and an integer $k \leq m$. |
| *Question:* | Is there an order $\pi = (\varphi_1, \ldots, \varphi_m)$ over $\Phi_+$ such that $\lvert\mathcal{K}(P) \cap S\mathcal{K}(P, \pi)\rvert \geq k$ holds? |

We start by establishing a general upper bound.

THEOREM 5.12. *For every efficiently computable JA rule $\mathcal{K}$ it holds that $S\mathcal{K}$-SUPPORTED-JUDGMENT is in NP.*

The omitted proof relies on Lemma 5.2. For the class of sequential quota rules we provide a matching lower bound by adapting the proof of Theorem 5.4.

THEOREM 5.13. *For every quota rule $F_q \in \mathcal{F}$ it holds that $SF_q$-SUPPORTED-JUDGMENT is NP-complete.*

Lastly, we highlight the significance of Lemma 5.2 for building a connection between our sequential rules and distance based rules. While it is not directly obvious, for $q = 1/2$, $SF_q$SJ is related to the maxcard subagenda rule as studied by Lang and Slavkovik [16]. In general, $S\mathcal{K}$SJ coincides with asking whether there exists a complete and consistent judgment $J \in \mathcal{J}(\Phi)$, such that $h(\mathcal{K}(P), J) \leq m - k$ (where $h(\mathcal{K}(P), J)$ denotes the hamming distance between $\mathcal{K}(P)$ and $J$). If there exists such an order $\pi$, for the resulting outcome $S\mathcal{K}(P, \pi)$ it clearly holds that $h(\mathcal{K}(P), S\mathcal{K}(P, \pi)) \leq m-k$. Vice versa, if there exists a judgment $J \in \mathcal{J}(\Phi)$ with $h(\mathcal{K}(P), J) \leq m-k$, we construct a valid order $\pi$ following Lemma 5.2 by arbitrarily positioning the supported formulas at the beginning. These observations may be an interesting tool for further research on computational complexity for counting problems.

# 6 SEQUENTIAL RULES AND THE MAXIMUM SUBAGENDA RULE

In this section we describe how we can link the sequential JA rules that we've studied to other well-known majority preserving JA rules. Particularly, we highlight the case with the majority rule as underlying rule to our sequential procedure. Hereby, we show that the maximum subagenda rule[3] (MSA), as defined by Lang and Slavkovik [16], exactly outputs the set of aggregated judgments which can also be derived by the sequential majority rule with suitable processing orders applied. This connection enables us to transfer some of our complexity results to related non-sequential procedures. In order to make the most out of this connection, we slightly generalize the MSA rule defined in [16] as described afterwards.

*Definition 6.1 (Generalized Maximum Subagenda Rule).* For an agenda $\Phi$ and a set $S \subseteq \Phi$ we define $max(S, \subseteq) \subset 2^S$ as the set consisting of inclusion maximal subsets of $S$ with respect to consistency. More formally, for $S' \subseteq S$ it holds that $S' \in max(S, \subseteq)$ if and only if $S'$ is consistent and there exists no consistent set

[3] Also known in JA as maximal Condorcet rule (see Lang et al. [15]), while the outcome is also denoted as Condorcet admissible set (see Nehring et al. [19]).

$S'' \subseteq S$ with $S' \subset S''$. For any complete and resolute JA rule $\mathcal{K}$, we define the (irresolute) **generalized maximum subagenda rule** $MSA_{\mathcal{K}}\colon \mathcal{J}(\Phi)^r \to 2^{\mathcal{J}(\Phi)}$ as follows. Let $P \in \mathcal{J}(\Phi)^r$ be a profile of judgments and $J \in \mathcal{J}(\Phi)$ a judgment, then $J \in MSA_{\mathcal{K}}(P)$ holds if and only if there exists a set $S \in max(\mathcal{K}(P), \subseteq)$ with $S \subseteq J$.

The MSA rule is irresolute, i.e., it returns a set of judgments as result, and equals the definition presented by Lang and Slavkovik [16] for $\mathcal{K} = F_{1/2}$. Having the MSA rule defined, we make the subsequent observation, establishing a connection between the MSA rule and our earlier studied sequential quota JA rules.

Theorem 6.2. *Let $P \in \mathcal{J}(\Phi)^r$ be a profile and $J \in \mathcal{J}(\Phi)$ a complete and consistent judgment. Then, $J \in MSA_{\mathcal{K}}(P)$ holds if and only if there exists an order $\pi$ over $\Phi_+$ with $S\mathcal{K}(P, \pi) = J$.*

Proof. We begin with the direction from left to right. By definition, $MSA_{\mathcal{K}}(P)$ contains every complete and consistent judgment $J$, such that there doesn't exist a consistent set $K \subseteq \mathcal{K}(P)$ satisfying $J \cap \mathcal{K}(P) \subset K$. Note that this especially holds for $|K| = |J \cap \mathcal{K}(P)| + 1$, i.e., $J \cap \mathcal{K}(P)$ cannot be extended by a single formula from $\mathcal{K}(P)$. Due to consistency of $J$ there is a satisfying truth assignment for $J \cap \mathcal{K}(P)$. Yet, no such truth assignment satisfies any formula in $\mathcal{K}(P) \setminus J$ and must thus satisfy its complement. Hence, it holds that $J \cap \mathcal{K}(P)$ must entail $J \setminus \mathcal{K}(P)$. Now, following a similar argumentation as in Lemma 5.2, for $\pi = (J \cap \mathcal{K}(P), J \setminus \mathcal{K}(P))$ we obtain $S\mathcal{K}(P, \pi) = J$ and therefore, the right side holds, too.

For the direction from right to left assume that there is an outcome $J = S\mathcal{K}(P, \pi)$ with $J \notin MSA_{\mathcal{K}}(P)$. Note that $J$ is consistent by definition and hence, its intersection with $\mathcal{K}(P)$ is consistent, too. By assumption, $J \cap \mathcal{K}(P)$ cannot be inclusion maximal in $\mathcal{K}(P)$ with respect to consistency as otherwise $J \in MSA_{\mathcal{K}}(P)$ would follow. Therefore, let $K \in max(\mathcal{K}(P), \subseteq)$, such that $J \cap \mathcal{K}(P) \subset K \subseteq \mathcal{K}(P)$ holds. Now, we construct an order $\pi'$ where $J \cap \mathcal{K}(P)$ is at the beginning of $\pi'$, immediately followed by $K \setminus J \cap \mathcal{K}(P)$, and all remaining formulas afterwards. With Lemma 5.2 it holds that $J = S\mathcal{K}(P, \pi')$ is true. Yet, $K \subseteq S\mathcal{K}(P, \pi')$ holds as well since $K$ is a consistent subset of $\mathcal{K}(P)$ processed at the beginning of $\pi'$. Hence we conclude that $K \subseteq J$ must hold, which is a contradiction to $J \cap \mathcal{K}(P) \subset K \subseteq \mathcal{K}(P)$. □

The previous theorem can be applied to transfer complexity results for our decision problems in Section 5. For complete and resolute JA rules $\mathcal{K}$, asking whether there exists an order $\pi$, such that some condition on the output $S\mathcal{K}(P, \pi)$ is satisfied, coincides with asking whether there is a judgment $J \in MSA_{\mathcal{K}}(P)$ satisfying the same condition. In particular, for $F_q = 1/2$ and a single formula $\varphi$ the problem $SF_q$-Winner-Design coincides with the existential MSA-Winner problem, while $SF_q$-Winner-Robustness coincides with the universal variant.[4]

This observation has multiple consequences. First of all, Lang and Slavkovik [16] showed the universal MSA-Winner problem is $\Pi_2^p$-complete, which aligns with our result from Corollary 5.10. However, the referenced result by Lang and Slavkovik requires a linear number of judges while two judges are sufficient for our proof. Consequently, our proof allows a stricter result than the one by Lang and Slavkovik. On the other hand our results also hold if we do not restrict MSA to the majority rule as underlying JA rule. In particular, upper bounds hold for every complete, efficiently computable, resolute rule, while hardness results hold for every of our quota rules.

The following corollaries follow from Theorems 5.7, 5.9 and 6.2, and only refer to existential problems, which imply related $\Pi_2^p$ results for the universal variants, by additionally following Lemma 5.6.

Corollary 6.3. *For any complete, efficiently computable, resolute JA rule $\mathcal{K}$ it holds that $MSA_{\mathcal{K}}$-Winner is in $\Sigma_2^p$.*

Corollary 6.4. *For every quota rule $F_q \in \mathcal{F}$ and even a constant number of judges it holds that $MSA_{F_q}$-Winner is $\Sigma_2^p$-complete.*

We explicitly highlight that the previous corollary holds for $q = 1/2$, and thereby enhances previous results on MSA.

## 7 CONCLUSION

We introduced the complexity theoretic study of problems related to sequential JA rules with a special focus on quota rules as the underlying rule. Our results are summarized in Table 1. We obtained completeness for a number of different complexity classes which show that the problems differ substantially even though they are very related. The study of sequential rules is very important since they model real-world decision making. To ensure consistency with the already decided formulas, it is important to solve the winner problem. On the other hand, we studied the manipulative power a designer of such a procedure possesses. The increase in complexity for the case where a single formula is the desired set indicates that the problem is actually harder than winner determination itself. As a task for future research other problems related to sequential JA rules have to be studied. Our study was mostly limited to the class of quota rules as underlying procedures and this should obviously be extended to more diverse underlying rules. De Haan [3] follows an approach to identify new ways of representing agendas via specific boolean formulas, such that the complexity of various problems related to JA becomes tractable, when the agenda is represented in a more limited way. Furthermore, he formulated the determination of the complexity of the winner problem for until yet unconsidered JA rules, which he hasn't studied, as future work. In a second step, the author suggests that one can use the tractable languages identified in his paper to study whether the complexity of the problems for the newly investigated JA rules can be decreased. Within our paper we have done the first part and determined the complexity of the winner problem for complete and consistent sequential JA rules. As future work we like to study how the tractable languages as defined by de Haan [3] affect our complexity results and possibly could even enable lower bounds. These results, when enabling tractability, might have enormous impact on the practical usage of the sequential JA rules we studied, since they are used in various scenarios and situations, as described earlier.

## ACKNOWLEDGMENTS

We thank all reviewers for their helpful comments. This work was supported by DFG grants BA 6270/1-1 and RO 1202/21-1 and by a grant from the NRW Ministry for Innovation, Science, and Research.

[4] Slightly abusing notation, we consider existential ($\exists J \in MSA_{\mathcal{K}}(P) : \{\varphi\} \subseteq J$) and universal variants of MSA$_{\mathcal{K}}$-Winner ($\forall J \in MSA_{\mathcal{K}}(P) : \{\varphi\} \subseteq J$) for irresolute rules.

## REFERENCES


[1] Dorothea Baumeister, Gábor Erdélyi, Olivia J. Erdélyi, and Jörg Rothe. 2015. Complexity of Manipulation and Bribery in Judgment Aggregation for Uniform Premise-Based Quota Rules. *Mathematical Social Sciences* 76 (2015), 19–30.

[2] Stephen A. Cook. 1971. The Complexity of Theorem-Proving Procedures. In *Proceedings of the 3rd ACM Symposium on Theory of Computing*. ACM Press, 151–158.

[3] Ronald de Haan. 2018. Hunting for Tractable Languages for Judgment Aggregation. In *Sixteenth International Conference on Principles of Knowledge Representation and Reasoning*.

[4] Ronald de Haan and Marija Slavkovik. 2017. Complexity Results for Aggregating Judgments using Scoring or Distance-Based Procedures. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 952–961.

[5] Franz Dietrich and Christian List. 2007. Judgment Aggregation by Quota Rules: Majority Voting Generalized. *Journal of Theoretical Politics* 19, 4 (2007), 391–424.

[6] Ulle Endriss. 2016. Judgment Aggregation. In *Handbook of Computational Social Choice*, Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia (Eds.). Cambridge University Press, Chapter 17, 399–426.

[7] Ulle Endriss. 2018. Judgment Aggregation with Rationality and Feasibility Constraints. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 946–954.

[8] Ulle Endriss and Ronald de Haan. 2015. Complexity of the Winner Determination Problem in Judgment Aggregation: Kemeny, Slater, Tideman, Young. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 117–125.

[9] Ulle Endriss, Umberto Grandi, Ronald De Haan, and Jérôme Lang. 2016. Succinctness of languages for judgment aggregation. (2016).

[10] Ulle Endriss, Umberto Grandi, and Daniele Porello. 2012. Complexity of Judgment Aggregation. *Journal of Artificial Intelligence Research* 45 (2012), 481–514.

[11] André Große, Jörg Rothe, and Gerd Wechsung. 2006. On Computing the Smallest Four-Coloring of Planar Graphs and Non-Self-Reducible Sets in P. *Information processing letters* 99, 6 (2006), 215–221.

[12] Wojciech Jamroga and Marija Slavkovik. 2013. Some Complexity Results for Distance-Based Judgment Aggregation. In *AI 2013: Advances in Artificial Intelligence - 26th Australasian Joint Conference, Dunedin, New Zealand, December 1-6, 2013. Proceedings*. 313–325.

[13] Lewis A. Kornhauser and Lawrence G. Sager. 1986. Unpacking the Court. *Yale Law Journal* 96, 1 (1986), 82–117.

[14] Mark W. Krentel. 1986. The Complexity of Optimization Problems. In *Proceedings of the eighteenth annual ACM symposium on Theory of computing*. 69–76.

[15] Jérôme Lang, Gabriella Pigozzi, Marija Slavkovik, Leendert van der Torre, and Srdjan Vesic. 2017. A Partial Taxonomy of Judgment Aggregation Rules and their Properties. *Social Choice and Welfare* 48, 2 (2017), 327–356.

[16] Jérôme Lang and Marija Slavkovik. 2014. How Hard is it to Compute Majority-Preserving Judgment Aggregation Rules?. In *Proceedings of the Twenty-first European Conference on Artificial Intelligence*. IOS Press, 501–506.

[17] Christian List. 2004. A Model of Path-Dependence in Decisions over Multiple Propositions. *American Political Science Review* 98, 3 (2004), 495–513.

[18] Michael K. Miller and Daniel Osherson. 2009. Methods for Distance-Based Judgment Aggregation. *Social Choice and Welfare* 32, 4 (2009), 575.

[19] Klaus Nehring, Marcus Pivato, and Clemens Puppe. 2014. The Condorcet set: Majority Voting over Interconnected Propositions. *Journal of Economic Theory* 151 (2014), 268–303.

[20] Christos Papadimitriou. 1994. *Computational Complexity*. Addison-Wesley.

[21] Bezalel Peleg and Shmuel Zamir. 2018. Judgments Aggregation by a Sequential Majority Procedure. *Mathematical Social Sciences* 95 (2018), 37–46.

[22] Daniele Porello and Ulle Endriss. 2014. Ontology Merging as Social Choice: Judgment Aggregation under the Open World Assumption. *Journal of Logic and Computation* 24, 6 (2014), 1229–1249.

[23] Christopher Umans. 1999. Hardness of Approximating $\Sigma_2^P$ Minimization Problems. In *40th Annual Symposium on Foundations of Computer Science (Cat. No. 99CB37039)*. IEEE, 465–474.

# 4 The Possible Winner with Uncertain Weights Problem

## 4.1 Summary

In this work we introduced a novel variant of the possible winner problem that we call the possible winner with uncertain weights problem. The original possible winner problem, first studied by Konczak and Lang [98], is a central problem of interest within and beyond the research area of preference aggregation by voting [102, 131]. Based on this thoroughly studied problem, we introduced a novel concept where an election's uncertainty no longer stems from partial preferences but from unknown weights for some of the votes. To formulate this new approach, we extended the original problem's elections from unweighted to weighted votes, i.e., assigning weights to the votes. Having done this, we formulated the possible winner with uncertain weights problem as follows for some voting rule $\mathcal{E}$ as well as $\mathbb{F} \in \{\mathbb{N}, \mathbb{Q}^+\}$.

| $\mathcal{E}$-Possible-Winner-with-Uncertain-Weights-$\mathbb{F}$ ($\mathcal{E}$-PWUW-$\mathbb{F}$) | |
|---|---|
| **Given:** | A list of candidates $C$, a list of unit-weight votes $V_1$ over $C$, a list of votes $V_0$ over $C$ with unspecified weights, and a designated candidate $c \in C$. |
| **Question:** | Are there weights $w_i \in \mathbb{F}$, $1 \leq i \leq \lvert V_0 \rvert$, for the votes in $V_0$, such that $c$ is a winner of the weighted $\mathcal{E}$ election $(C, V_1 \cup V_0)$? |

Based on this generic problem variant we introduced three further sub-variants, namely

1. $\mathcal{E}$-PWUW-rw-$\mathbb{F}$: extending the initial variant by ranges $R_i = [\ell_i, r_i] \subseteq \mathbb{F}$ for all votes $v_i \in V_0$ from which weights for the votes must be chosen,

2. $\mathcal{E}$-PWUW-bw-$\mathbb{F}$: extending the initial variant by some upper bound $B \in \mathbb{F}$ limiting the sum of all weights distributed to votes in $V_0$, and

3. $\mathcal{E}$-PWUW-bw-rw-$\mathbb{F}$: combining the two previous variants into one aggregated one with both requirements, ranges $R_i$ and an upper bound $B$.

Besides introducing these problem variants and discussing their relationships to other election-related decision problems, such as the constructive control problem $\mathcal{E}$-Constructive-Control-by-Adding-Voters (CCAV), we studied the computational complexity of the introduced variants for nonnegative integer weights, i.e., when $\mathbb{F} = \mathbb{N}$, and nonnegative rational weights, i.e., when $\mathbb{F} = \mathbb{Q}^+$. Thereby, we establish exact computational complexity bounds for all the variants and several election rules such as $k$-approval, $k$-veto, plurality with runoff, veto with runoff, Borda, simplified Bucklin and fallback voting, Copeland$^\alpha$, as well as ranked pairs. Our established computational bounds range from P-membership up to NP-completeness. For some results more preliminary work was required, and hence, among other auxiliary propositions, we also proved that for both voting rules, plurality with runoff and veto with runoff, the problem CCAV in succinct representation belongs to P.

## 4.2 Publication

D. Baumeister, M. Neveling, M. Roos, J. Rothe, L. Schend, R. Weishaupt, and L. Xia. "The Possible Winner with Uncertain Weights Problem". In: *Journal of Computer and System Sciences* (Submitted)

Preliminary versions of this work were submitted to and accepted at the *23rd International Symposium on Fundamentals of Computation Theory*, where this work was awarded the best paper award, and the *20th European Conference on Artificial Intelligence*:

M. Neveling, J. Rothe, and R. Weishaupt. "The Possible Winner Problem with Uncertain Weights Revisited". In: *Proceedings of the 23rd International Symposium on Fundamentals of Computation Theory*. Springer-Verlag LNCS, Sept. 2021, pp. 399–412

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*. IOS Press, Aug. 2012, pp. 133–138

## 4.3 Personal Contribution

The writing of this work was conducted jointly with my co-authors Dorothea Baumeister, Marc Neveling, Magnus Roos, Jörg Rothe, Lena Schend, and Lirong Xia. Initial versions of Propositions 5.4 and 5.5 as well as Theorems 3.4, 5.6, 5.10, 5.11, 5.12, and 5.18 were contributed by myself, whereas Theorem 5.13 and Theorem 5.16 are joint work with my co-author Marc Neveling.

# The Possible Winner with Uncertain Weights Problem*

Dorothea Baumeister[a], Marc Neveling[a], Magnus Roos[a], Jörg Rothe[a,*], Lena Schend[a], Robin Weishaupt[a], Lirong Xia[b]

[a] *Institut für Informatik, Heinrich-Heine-Universität Düsseldorf, 40225 Düsseldorf, Germany*
[b] *Computer Science Department, Rensselaer Polytechnic Institute, Troy, NY 12180-3590, USA*

**Abstract**

The original possible winner problem consists of an unweighted election with partial preferences and a distinguished candidate $c$ and asks whether the preferences can be extended to total ones such that $c$ wins the given election. We introduce a novel variant of this problem: possible winner with uncertain weights. In this variant, for a given weighted election, not some of the preferences but some of the preferences' weights are uncertain. We introduce a general framework to study this problem for nonnegative integer and rational weights as well as for four different variations of the problem itself: with and without given upper bounds on the total weight and with and without given ranges to choose weights from. We study the complexity of these problems for important voting systems such as scoring protocols, (simplified) Bucklin and fallback voting, plurality with runoff and veto with runoff, Copeland$^\alpha$, ranked pairs, and Borda.

*Keywords:* computational social choice, possible winner, voting rule, uncertainty, computational complexity

## 1. Introduction

Over the previous decades, computational social choice—with its many applications to collective decision making—has evolved into a central subarea of artificial intelligence and, in particular, multiagent systems. Looking into the textbook edited by Brandt et al. [17], one of the most intensively studied problems in computational social choice alongside manipulation, control, and bribery (see the surveys by Faliszewski et al. [31, 33]) is the *possible winner problem* that Konczak and Lang [45] were the first to study. Generalizing the (unweighted) coalitional manipulation problem [21] and being a special case of swap bribery [26], in this problem we are given an election with only partial instead of total preferences over the candidates, a distinguished candidate $c$, and we ask whether one can extend the partial preferences to total ones to make $c$ a winner. This problem and variations thereof as well as its companion, the *necessary winner*

*A preliminary version of this work appeared in the proceedings of the *20th European Conference on Artificial Intelligence* (ECAI 2012) [6] and of the *23rd International Symposium on Fundamentals of Computation Theory* (FCT 2021) [52].

*Corresponding author

*problem* [45, 58] and its variations, have been studied by many authors for many voting rules—see, e.g., the recent survey by Lang [47] and the references cited therein.

The idea underlying the possible and necessary winner problems (to determine the winners in the presence of incomplete preferences for *some* or for *all* extensions to total preferences) is so fundamental that it has been applied successfully to many other areas, including *judgment aggregation* [7, 10], *fair division* [8, 46], *hedonic games* [44], and *abstract argumentation* [9, 11, 53, 57]. Betzler and Dorn [14] and Baumeister and Rothe [3] established a dichotomy result for the possible winner problem, Betzler et al. [13, 15] investigated the parameterized complexity of this problem, and Bachrach, Betzler, and Faliszewski [1] investigated a probabilistic variant thereof. Chevaleyre et al. [19] introduced the *possible winner with respect to the addition of new alternatives problem*, which is related to, yet different from the *problem of control by adding candidates*[1] [2, 42] and is also similar, yet not identical to the *cloning problem* in elections [27]. Their variant was further studied, e.g., by Xia, Lang, and Monnot [59] (see their joint journal version [20]) and by Baumeister, Roos, and Rothe [4]. The latter paper was the first to consider a weighted variant of the possible winner problem, and it also introduced and studied this problem under voting rule uncertainty, an approach that was followed up by Elkind and Erdélyi [25], who applied it to coalitional manipulation [21], and by Baumeister and Hogrebe [12]. Baumeister et al. [5] studied variants of the possible winner problem with truncated ballots. While most work on the possible winner problem is concerned with an unweighted variant of the problem, Baumeister et al. [4] were the first to consider its *weighted* variant.

We present a general framework to study the *weighted* possible winner problem, and we focus on elections where not some of the voters' preferences, but some of their *weights* are uncertain. The problems we study in our framework come with nonnegative integer or rational weights, with or without given upper bounds on the total weight to be assigned, and with or without given ranges to choose the weights from. An interesting point with this focus is that while the original possible winner problem generalizes the coalitional manipulation problem [21], certain variants of the possible winner with uncertain weights problem that we study generalize constructive control by adding or deleting voters [2, 42]. We study the resulting problem variants in terms of their computational complexity for various central and natural voting systems: scoring protocols, (simplified) Bucklin and fallback voting, plurality with runoff and veto with runoff, Copeland$^{\alpha}$, ranked pairs, and Borda.

Based on the ever-increasing exchange of—almost real-time—data in our modern society, the possible winner with uncertain weights model can be applied to all kinds of elections today better than ever. For almost every election taking place, some pre-election polls are done and the results are published. These results can be translated by suitable probability-based methods into ranges for likely weights, upper bounds on the total weight, etc., so that the results presented here provide powerful means to improve election forecasts in efficient ways.

The following situation provides an additional, motivational example why it is interesting to study the possible winner with uncertain weights problem. Imagine a com-

[1] The terms *candidate* and *alternative* are used synonymously.

pany that is going to decide on its future strategy by voting at the annual general assembly of stockholders. Among the parties involved, everybody's preferences are common knowledge. However, who will succeed with its preferred alternative for the future company strategy depends on the stockholders' weights, i.e., on how many stocks they each own, and there is uncertainty about these weights. Is it possible to assign weights to the parties involved (e.g., by buying new stocks) such that a given alternative wins?

Our work is structured as follows. Section 2 formally introduces all required preliminaries, i.e., notation, notions, as well as the voting systems that we will study in the subsequent sections. In Section 3, we introduce the possible winner with uncertain weights problem and all four variants that we study. Furthermore, we establish general results with respect to these problems, highlight and discuss links to other related decision problems, and formulate preprocessing steps that simplify our ensuing proofs. In Section 4, we study the possible winner with uncertain weights problem for nonnegative rational weights (i.e., weights in $\mathbb{Q}^+$). Section 5 covers our results for the possible winner with uncertain weights problem for nonnegative integer weights (i.e., weights in $\mathbb{N}$) and is separated into several subsections, each covering all results for specific voting systems introduced in the preliminaries. In particular, Section 5.3 covers our results for plurality with runoff (and also proves that constructive control by adding voters in succinct representation belongs to P) while Section 5.4 proves an analogous result for veto with runoff. Finally, we summarize our findings in Section 6, provide an overview of our results in Table 7 for rational weights and in Table 8 for integer weights, and discuss possible future work.

## 2. Preliminaries

An *election* is a pair $(C,V)$ consisting of a finite set $C$ of *candidates* and a finite list $V$ of *votes* representing the voters' preferences over the candidates in $C$ and occasionally denoted by $v_1,\dots,v_n$ when there are $n=|V|$ votes. We assume that each vote is a (strict) linear preference order. For example, if there are four candidates in $C=\{a,b,c,d\}$ and a voter prefers $b$ to $c$, $c$ to $d$, and $d$ to $a$, we write this vote as $b>c>d>a$. If such an order is not total (e.g., when a voter only specifies $a>c>d$ as her preference over these four candidates), we say it is a *partial order*. For winner determination in weighted elections, a vote $v$ of weight $w$ is considered as if there were $w$ unweighted (i.e., unit-weight) votes $v$. When possible, we represent elections *succinctly*, i.e., identical votes are not listed one by one but just once along with a number in binary representation giving the multiplicity of this vote.

For a given election $(C,V)$, the *weighted majority graph* (WMG) is defined as a directed graph whose vertices are the candidates, and we have an edge $(c,d)$ of weight $N(c,d)$ between any two vertices $c$ and $d$, where $N(c,d)$ is the number of voters preferring $c$ to $d$ minus the number of voters preferring $d$ to $c$. Note that in the WMG of any election, all weights on the edges have the same parity (and whether it is odd or even depends on the parity of the number of voters), and $N(c,d)=-N(d,c)$ (which is why it is enough to give only one of these two edges explicitly—the edge with nonnegative weight; in case of a tie $N(c,d)=N(d,c)=0$ for an even number of voters, this edge is undirected).

A *voting rule* determines the winner(s) of a given election. An important class of voting rules are the *scoring protocols*: For $m$ candidates, a *scoring vector* $\vec{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_m)$ of integers with $\alpha_1 \geq \alpha_2 \geq \cdots \geq \alpha_m$ specifies the points the candidates receive from each voter based on their position in it, i.e., a candidate in the $i$-th position of a vote receives $\alpha_i$ points from it, and the *score of candidate $c \in C$ in election* $(C,V)$, denoted by $score_V(c)$, is the sum of the points $c$ receives from the votes in $V$. Formally, denoting the position of candidate $c$ in vote $v_j$ by $\rho_j(c)$, $c$ receives $\alpha_{\rho_j(c)}$ points from $v_j$, and the total score of $c$ can be written as $score_V(c) = \sum_{j=1}^{n} \alpha_{\rho_j(c)}$ for $n$ votes. All candidates with largest score are denoted as the $\vec{\alpha}$ winners. Furthermore, we say that a scoring protocol $\mathscr{E}$ is a *binary scoring protocol* if all entries of its scoring vector are from $\{0,1\}$. Specifically, we consider the prominent scoring protocols (for $m$ candidates) that are based on the following scoring vectors:

**k-approval:** $(1,\ldots,1,0,\ldots,0)$, where the first $k \leq m$ entries are ones (1-approval is also known as *plurality*);

**k-veto:** $(1,\ldots,1,0,\ldots,0)$, where the last $k \leq m$ entries are zeros (1-veto is also known as *veto*); and

**Borda:** $(m-1, m-2, \ldots, 0)$.

For example, $k$-veto and $k$-approval are binary scoring protocols for all $k \in \mathbb{N}$ and any number $m \geq k$ of candidates, whereas Borda is a binary scoring protocol only for $m \leq 2$ candidates.

In addition to scoring protocols, we consider the following voting rules:

**Copeland$^\alpha$ (for each rational number $\alpha$, $0 \leq \alpha \leq 1$):**[2] For any two candidates $c$ and $c'$, we run a pairwise contest between them, by seeing how many voters prefer $c$ to $c'$, and how many prefer $c'$ to $c$; the winner of the pairwise contest is the one preferred more often. Each candidate receives one point for each win in a pairwise contest, $\alpha$ points for each tie, and zero points for each loss. This is the Copeland$^\alpha$ score of the candidate and a Copeland$^\alpha$ winner maximizes the Copeland$^\alpha$ score.

**Ranked pairs:** This rule first creates an entire ranking of all the candidates. In each step, we consider a pair of candidates, $c$ and $c'$, that we have not yet considered previously; specifically, we choose among the remaining pairs one with the highest $N(c,c')$ value (in case of ties, we use some tie-breaking mechanism) and then fix the order $c > c'$, unless this contradicts previous orders already fixed (i.e., unless this order violates transitivity). We continue until we have considered all pairs of candidates (and so we have a full ranking). The candidate at the top of the ranking wins, for the chosen tie-breaking mechanism.

**Plurality with runoff** proceeds in two stages. In the first stage, all candidates except the two candidates with the highest and the second-highest plurality score are

[2] The original Copeland system [22] is defined for the specific value of $\alpha = 1/2$; the generalization to other $\alpha$ values is due to Faliszewski et al. [32].

eliminated. In the second stage (the runoff), among the two remaining candidates and with votes restricted to these, the candidate with the highest plurality score wins. In both stages, we use some predefined tie-breaking rule to determine the two candidates that proceed to the runoff and the overall winner in case there are ties.

**Veto with runoff** works just like plurality with runoff, except it uses veto scores in both stages to determine who proceeds to the runoff and who is the overall winner.

**Bucklin (BV) and fallback voting (FV), both simplified:** In a Bucklin election, the voters' preferences are linear orders and the level $\ell$ score of a candidate $c$ is the number of voters ranking $c$ among their top $\ell$ positions. The Bucklin score of a candidate $c$ is the smallest number $t$ such that more than half of the voters rank $c$ somewhere in their top $t$ positions. A Bucklin winner minimizes the Bucklin score.[3] In (simplified) fallback elections, on the other hand, nontotal preference orders are allowed. Every Bucklin winner is also a fallback winner, but if no Bucklin winner exists (which may happen due to the voters' partial orders) and $\ell$ is the length of a longest preference order among the votes, all candidates with the greatest level $\ell$ score are the fallback winners. Throughout this paper we will refer to "simplified Bucklin" and "simplified fallback" simply as Bucklin and fallback voting.

We consider the *nonunique-winner model*, which means that it is enough for the distinguished candidate to be one among possibly several candidates that win the election.

In our proofs, we make use of the following notation. Let $S \subseteq C$ be a set of candidates. When $\overrightarrow{S}$ appears in a vote, the candidates from $S$ are ranked in any fixed (e.g., the lexicographical) order; when $\overleftarrow{S}$ appears in a vote, the candidates from $S$ are ranked in the reverse order; when $S$ appears in a vote (without an arrow on top), the order in which the candidates from $S$ are ranked here does not matter for our argument; and when $\cdots$ appears in a vote, the order in which the remaining candidates occur does not matter. For example, if $C = \{a,b,c,d\}$, $S = \{b,c\}$, and we use a lexicographical order, then $\overrightarrow{S} > d > a$ means $b > c > d > a$; $\overleftarrow{S} > d > a$ means $c > b > d > a$; and both $S > d > a$ and $\cdots > d > a$ indicate any one of $b > c > d > a$ and $c > b > d > a$. Further, when we consider a scoring protocol or plurality with runoff or veto with runoff for an election $(C,V)$ and two candidates $c,d \in C$, we use

$$\mathit{diff}_{(C,V)}(c,d) = \mathit{score}_V(c) - \mathit{score}_V(d)$$

to denote the difference of the scores of $c$ and $d$ in $(C,V)$. By $\mathit{diff}_{(\{c,d\},V)}(c,d)$ we denote the difference of the scores of $c$ and $d$ in the head-to-head contest which is $(\{c,d\},V)$ with the votes in $V$ being tacitly reduced to only $c$ and $d$.

[3] We consider only this simplified version of Bucklin voting. In the full version (see, e.g., [28, 29]), among all candidates with smallest Bucklin score, say $t$, for $c$ to win it is also required that $c$'s level $t$ score is largest.

Some of our proofs will use *McGarvey's trick* [50] (applied to WMGs), which constructs a list of votes whose WMG is the same as some targeted weighted directed graph. This is helpful to present our proofs, as we only need to specify the WMG instead of the whole list of votes, and then by using McGarvey's trick for WMGs, an appropriate vote list can be constructed in polynomial time. More specifically, McGarvey showed that for every unweighted majority graph, there is a particular list of preferences that results in this majority graph. Extending this to WMGs, the trick works as follows. For any pair of candidates, $(c,d)$, if we add two votes, $c > d > c_3 > \cdots > c_m$ and $c_m > c_{m-1} > \cdots > c_3 > c > d$, to a vote list, then in the WMG, the weight on the edge $(c,d)$ is increased by 2 and the weight on the edge $(d,c)$ is decreased by 2, while the weights on all other edges remain unchanged.

We assume the reader to be familiar with the basic concepts of computational complexity, such as the complexity classes P and NP and the notions of NP-*hardness* and NP-*completeness*, based on *polynomial-time many-one reductions*, denoted by $A \leq_{\mathrm{m}}^{\mathrm{p}} B$. For more background on complexity theory, we refer to the textbooks by Garey and Johnson [38], Papadimitriou [54], and Rothe [55]. In our proofs, we also provide polynomial-time many-one reductions to the NP-complete problem EXACT-COVER-BY-3-SETS, see [38], which is defined as follows:

| EXACT-COVER-BY-3-SETS (X3C) | |
|---|---|
| **Given:** | Given a set $\mathscr{B} = \{b_1, \ldots, b_{3q}\}$ and a collection $\mathscr{S} = \{S_1, \ldots, S_n\}$ with $\lvert S_i \rvert = 3$ and $S_i \subseteq \mathscr{B}$ for $1 \leq i \leq n$. |
| **Question:** | Is there a subset $\mathscr{S}' \subseteq \mathscr{S}$ such that every element of $\mathscr{B}$ occurs in exactly one member of $\mathscr{S}'$? |

## 3. The Possible Winner With Uncertain Weights Problem

With all required preliminaries defined in the previous section, in this section we formally introduce the possible winner with uncertain weights problem as well as three further variants of the problem that we study. Furthermore, we will discuss our decisions as to why we specify our problem variants the way we do, and we will provide connections to familiar problems. Besides that, we also formulate and prove some useful general results.

For a given voting system $\mathscr{E}$ and for $\mathbb{F} \in \{\mathbb{Q}^+, \mathbb{N}\}$, we define the general *possible winner with uncertain weights problem* as follows:

| $\mathscr{E}$-POSSIBLE-WINNER-WITH-UNCERTAIN-WEIGHTS-$\mathbb{F}$ ($\mathscr{E}$-PWUW-$\mathbb{F}$) | |
|---|---|
| **Given:** | A list of candidates $C$, a list of unit-weight votes $V_1$ over $C$, a list of votes $V_0$ over $C$ with unspecified weights, and a distinguished candidate $c \in C$. |
| **Question:** | Are there weights $w_i \in \mathbb{F}$, $1 \leq i \leq \lvert V_0 \rvert$, for the votes in $V_0$ such that $c$ is a winner of the weighted $\mathscr{E}$ election $(C, V_1 \cup V_0)$? |

We distinguish between allowing nonnegative rational weights, i.e., weights in $\mathbb{Q}^+$, and nonnegative integer weights, i.e., weights in $\mathbb{N}$. In particular, we allow weight-zero votes in $V_0$. Note that for inputs with $V_0 = \emptyset$ (which is not excluded in the problem definition), we obtain the ordinary unweighted (i.e., unit-weight) winner problem for $\mathscr{E}$.

Allowing weight zero for votes in $V_0$ in some sense corresponds to *control by deleting voters* [2, 42]. The reason why we distinguish between votes with uncertain weights and unit-weight votes in our problem instances is that we want to capture these problems in their full generality, just as votes with total preferences are allowed to occur in the instances of the original possible winner problem. The requirement of normalizing the weights in $V_1$ to unit-weight, on the other hand, *is* a restriction that is chosen on purpose and will help simplify some of the upcoming proofs. Instances that do not satisfy this requirement can simply be transformed to ones that do. To do so, we can duplicate votes from $V_1$ of weight greater than 1 according to their weight and add these duplicates as well as the original vote back to $V_1$, each with weight 1. Alternatively, if the resulting instance is no longer polynomial in the size of the original instance, we can interpret it as an instance in succinct representation and see the weights of the votes in $V_1$ as their multiplicities.

Besides the general $\mathscr{E}$-PWUW-$\mathbb{F}$ problem, we also consider the following three variants with additional restrictions:

1. $\mathscr{E}$-PWUW-RW-$\mathbb{F}$: In this variant, an $\mathscr{E}$-PWUW-$\mathbb{F}$ instance and regions (i.e., intervals) $R_i \subseteq \mathbb{F}$, $1 \leq i \leq |V_0|$, are given, and the question is the same as in $\mathscr{E}$-PWUW-$\mathbb{F}$, except that each weight $w_i$ must be chosen from $R_i$ in addition.
2. $\mathscr{E}$-PWUW-BW-$\mathbb{F}$: In this variant, an $\mathscr{E}$-PWUW-$\mathbb{F}$ instance and a positive bound $B \in \mathbb{F}$ is given, and the question is the same as in $\mathscr{E}$-PWUW-$\mathbb{F}$, except that $\sum_{i=1}^{|V_0|} w_i \leq B$ must hold in addition. In other words, the total weight that can be assigned to votes from $V_0$ is bounded by $B$.
3. $\mathscr{E}$-PWUW-BW-RW-$\mathbb{F}$: In this variant, an $\mathscr{E}$-PWUW-BW-$\mathbb{F}$ instance and regions $R_i \subseteq \mathbb{F}$, $1 \leq i \leq |V_0|$, are given, and the question is the same as in $\mathscr{E}$-PWUW-BW-$\mathbb{F}$, except that each weight $w_i$ must be chosen from its region $R_i$ in addition.

As already mentioned in the preliminaries, we focus on the nonunique-winner model, so the question always is whether $c$ can be made $a$ winner by assigning appropriate weights. However, by minor proof adjustments, most of our results can be shown to also hold in the *unique-winner* model where one asks whether $c$ can be made *the* only winner.

Let us now provide some general results that are useful for our further results. First of all, we provide some trivial reductions between the problem variants that we just have introduced.

**Theorem 3.1.** *For any voting rule $\mathscr{E}$ and $\mathbb{F} \in \{\mathbb{Q}^+, \mathbb{N}\}$, we have*

$$\mathscr{E}\text{-PWUW-RW-}\mathbb{F} \leq_{\mathrm{m}}^{\mathrm{p}} \mathscr{E}\text{-PWUW-BW-RW-}\mathbb{F} \quad \textit{and}$$

$$\mathscr{E}\text{-PWUW-BW-}\mathbb{F} \leq_{\mathrm{m}}^{\mathrm{p}} \mathscr{E}\text{-PWUW-BW-RW-}\mathbb{F}.$$

**Proof.** For the first reduction, let $I = (C, V_1, V_0, c, R)$ be an $\mathscr{E}$-PWUW-RW-$\mathbb{F}$ instance with $R = \{[l_i, r_i] \subseteq \mathbb{F} \mid v_i \in V_0\}$. To construct an equivalent $\mathscr{E}$-PWUW-BW-RW-$\mathbb{F}$ instance $I'$, we copy the instance $I$ and, additionally, set $B = \sum_{i=1}^{|V_0|} r_i$. Obviously, $I'$ is a YES-instance of $\mathscr{E}$-PWUW-BW-RW-$\mathbb{F}$ if and only if $I$ is a YES-instance of $\mathscr{E}$-PWUW-RW-$\mathbb{F}$.

For the second reduction, let $I = (C, V_1, V_0, c, B)$ be an $\mathscr{E}$-PWUW-BW-$\mathbb{F}$ instance with $B \in \mathbb{F}$. To construct an equivalent $\mathscr{E}$-PWUW-BW-RW-$\mathbb{F}$ instance $I'$, we copy the instance $I$ and, additionally, define $R' = \{[0,B] \subseteq \mathbb{F} \mid v_i \in V_0\}$. Obviously, $I'$ is a YES-instance of $\mathscr{E}$-PWUW-BW-RW-$\mathbb{F}$ if and only if $I$ is a YES-instance of $\mathscr{E}$-PWUW-BW-$\mathbb{F}$. ❑

Related to our variants of the $\mathscr{E}$-PWUW-$\mathbb{F}$ problem is the problem of *constructive control by adding voters*, $\mathscr{E}$-CCAV for short, as defined by Bartholdi, Tovey, and Trick [2]:

| $\mathscr{E}$-CONSTRUCTIVE-CONTROL-BY-ADDING-VOTERS ($\mathscr{E}$-CCAV) | |
|---|---|
| **Given:** | A list of candidates $C$, a list of registered votes $V$ over $C$, a list of as yet unregistered votes $V'$ over $C$, a distinguished candidate $c \in C$, and a nonnegative integer $k \in \mathbb{N}$. |
| **Question:** | Is there a subset $W \subseteq V'$ of votes from $V'$ with $\|W\| \leq k$ such that $c$ is a winner of the modified $\mathscr{E}$ election $(C, V \cup W)$? |

There is a simple, direct polynomial-time many-one reduction from CCAV to PWUW-BW-RW-$\mathbb{N}$, as the next proposition shows.

**Proposition 3.2.** $\mathscr{E}$-CCAV $\leq^{\mathrm{p}}_{\mathrm{m}}$ $\mathscr{E}$-PWUW-BW-RW-$\mathbb{N}$.

**Proof.** Let $I = (C, V, V', c, k)$ be an $\mathscr{E}$-CCAV instance. Construct an equivalent $\mathscr{E}$-PWUW-BW-RW-$\mathbb{N}$ instance $I' = (C', V_1, V_0, c', B, R)$ by setting $C' = C$, $V_1 = V$, $V_0 = V'$, $B = k$, $R = \{\{0,1\} \mid v_i \in V'\}$, and $c' = c$. Obviously, $I'$ is a YES-instance of $\mathscr{E}$-PWUW-BW-RW-$\mathbb{N}$ if and only if $I$ is a YES-instance of $\mathscr{E}$-CCAV. ❑

Note that this reduction can easily be adjusted to also hold when we assume the $\mathscr{E}$-CCAV instance to be in succinct representation. In this case, we just need to alter the weights' regions such that they reflect the votes' frequencies in the CCAV instance. For example, if $v \in V'$ has a frequency of 5, we would add $v$ to $V_0$ and set its region to $[0,5]$ instead of $\{0,1\}$. Furthermore, when using succinct representation for $\mathscr{E}$-CCAV instances, we can also provide a polynomial-time many-one reduction in the opposite direction. To emphasize the difference between these two problems notationally, we write $\mathscr{E}$-CCAV$_{\mathrm{succ}}$ when referring to the problem whose instances are in succinct representation.

**Proposition 3.3.** $\mathscr{E}$-PWUW-BW-RW-$\mathbb{N}$ $\leq^{\mathrm{p}}_{\mathrm{m}}$ $\mathscr{E}$-CCAV$_{\mathrm{succ}}$.

**Proof.** Let $I = (C, V_1, V_0, c, B, R)$ be an $\mathscr{E}$-PWUW-BW-RW-$\mathbb{N}$ instance. We construct an $\mathscr{E}$-CCAV$_{\mathrm{succ}}$ instance $I' = (C', V, V', c', k)$ in succinct representation as follows. First, we set $C' = C$, $V = V_1$, and $c' = c$. Second, for every vote $v_i \in V_0$ with weight-restricting region $R_i = [l_i, r_i]$, we add $l_i$ copies of $v_i$ to $V$ and $r_i - l_i$ copies of $v_i$ to $V'$. Since the values $r_i$ and $l_i$ do not need to be polynomial in the size of $I$, we indeed require succinct representation for $I'$ here, to keep the size of $I'$ polynomial in the size of $I$. Finally, we define $k = B - \sum_{i=1}^{|V_0|} l_i$. This completes the construction of $I'$ and,

obviously, $I'$ is a YES-instance of $\mathscr{E}$-CCAV$_{\text{succ}}$ if and only if $I$ is a YES-instance of $\mathscr{E}$-PWUW-BW-RW-$\mathbb{N}$. ❑

Since there are reductions in both directions, complexity results immediately carry over from $\mathscr{E}$-CCAV$_{\text{succ}}$ to $\mathscr{E}$-PWUW-BW-RW-$\mathbb{N}$ when we assume succinct representation. This will be useful later on, and we will refer to this fact when stating the corresponding results derived from it. However, note that the NP-hardness results on CCAV for Bucklin and Fallback voting by Erdélyi et al. [28, 29] concern the full, not the simplified versions of these voting rules, so their results do not carry over to our problems. Although with this two-sided reduction we already obtain quite some complexity results, we still cover these cases again in our later proofs in the next sections, as our proofs handle several variants of the PWUW problems at the same time. Furthermore, as we have reductions in both directions, all our results for $\mathscr{E}$-PWUW-BW-RW-$\mathbb{N}$ in the next sections also carry over to $\mathscr{E}$-CCAV$_{\text{succ}}$ in succinct representation. However, we do not highlight this fact in all further sections but state this here only once, so that readers interested in $\mathscr{E}$-CCAV$_{\text{succ}}$ in succinct representation are aware of this and can use these results accordingly.

Fitzsimmons and Hemaspaandra [35] have comprehensively studied election problems in succinct representation, so-called *high-multiplicity election problems*. Since such compactly represented instances may be exponentially smaller than in standard representation, polynomial-time algorithms for election problems in standard representation can become exponential-time when instances are represented succinctly. However, Fitzsimmons and Hemaspaandra [35] were able to either adapt these algorithms accordingly or to design new algorithms showing that all these problems in fact remain solvable in polynomial time: They could not detect a single natural case where such a problem would not remain in P when switching to high-multiplicity representation. On the other hand, for winner determination in Kemeny elections, which in standard representation is known to be $\mathrm{P}^{\mathrm{NP[log]}}$-complete [41], Fitzsimmons and Hemaspaandra [35] show that the complexity raises to $\mathrm{P}^{\mathrm{NP}}$-completeness for high-multiplicity representation. They also explore the relationship between high-multiplicity scheduling and manipulating high-multiplicity elections.

Lastly, we provide some general results for scoring protocols. To begin, we show for all binary scoring protocols $\mathscr{E}$—recall, a scoring protocol is binary if all its scoring vector's entries are from $\{0,1\}$—that the problems $\mathscr{E}$-PWUW-RW-$\mathbb{N}$ and $\mathscr{E}$-PWUW-$\mathbb{N}$ can be solved in P.

**Theorem 3.4.** *For any binary scoring protocol $\mathscr{E}$, the problems $\mathscr{E}$-PWUW-RW-$\mathbb{N}$ and $\mathscr{E}$-PWUW-$\mathbb{N}$ are in* P.

**Proof.** We first prove that $\mathscr{E}$-PWUW-RW-$\mathbb{N}$ is in P, and afterwards we will provide a simple reduction from $\mathscr{E}$-PWUW-$\mathbb{N}$ to $\mathscr{E}$-PWUW-RW-$\mathbb{N}$ such that the second result follows immediately.

Let $I = (C, V_1, V_0, c, R)$ be an $\mathscr{E}$-PWUW-RW-$\mathbb{N}$ instance. Denote by $V' \subseteq V_0$ all votes from $V_0$ which assign a score of 1 to $c$ and, furthermore, denote by $C' = \{d \in C \mid score_{V_1}(d) > score_{V_1}(c)\}$ all candidates in $C$ obtaining a higher score than $c$ in $V_1$. To check whether $I$ is a YES-instance of $\mathscr{E}$-PWUW-RW-$\mathbb{N}$, we proceed as follows. For

every candidate $d \in C'$, we check whether there exist votes in $V'$ allocating 0 points to $d$ such that the sum of the corresponding interval's upper limits is at least $score_{V_1}(d) - score_{V_1}(c)$. If that is the case, we assign to every such vote $v_i$ its maximum possible weight $r_i$ for $R_i = [l_i, r_i]$. In that case we know that

$$\begin{aligned} & score_{V_0}(c) - score_{V_0}(d) \geq score_{V_1}(d) - score_{V_1}(c) \\ \Leftrightarrow \quad & score_{V_1}(c) + score_{V_0}(c) \geq score_{V_1}(d) + score_{V_0}(d), \end{aligned}$$

so $c$ is a winner of the election and, hence, $I$ indeed is a YES-instance of $\mathscr{E}$-PWUW-RW-$\mathbb{N}$. Obviously, if this condition is not satisfied for some candidate $d \in C \setminus \{c\}$, it follows that $I$ is a NO-instance of $\mathscr{E}$-PWUW-RW-$\mathbb{N}$, as $c$ cannot beat $d$. Consequently, we can solve $\mathscr{E}$-PWUW-RW-$\mathbb{N}$ in polynomial time by this approach.

Now, let $I = (C, V_1, V_0, c)$ be an $\mathscr{E}$-PWUW-$\mathbb{N}$ instance. We construct an equivalent $\mathscr{E}$-PWUW-RW-$\mathbb{N}$ instance $I' = (C, V_1, V_0, c, R)$ by setting $R = \{[0, |V_1|] \mid v \in V_0\}$. Assume $I$ to be a YES-instance of $\mathscr{E}$-PWUW-$\mathbb{N}$. Hence, there exist weights $w_i \in \mathbb{N}$ for the votes $v_i \in V_0$ such that $c$ is a winner of the weighted election $(C, V_1 \cup V_0)$. However, for all $d \in C$, it must hold that $|score_{V_1}(d) - score_{V_1}(c)| \leq |V_1|$, so we can assume $0 \leq w_i \leq |V_1|$ for all weights $w_i$. Hence, we can use the same weights for $I'$ and, therefore, $I'$ is a YES-instance of $\mathscr{E}$-PWUW-RW-$\mathbb{N}$ as well. On the other hand, if $I'$ is a YES-instance of $\mathscr{E}$-PWUW-RW-$\mathbb{N}$, it is straightforward that $I$ is a YES-instance of $\mathscr{E}$-PWUW-$\mathbb{N}$ as well. Therefore, we obtain $\mathscr{E}$-PWUW-$\mathbb{N} \leq_{\mathrm{m}}^{\mathrm{p}} \mathscr{E}$-PWUW-RW-$\mathbb{N}$, and consequently, $\mathscr{E}$-PWUW-$\mathbb{N}$ is in P, too. ❑

We will use Theorem 3.4 in later sections when we study binary scoring protocols in more depth, to argue that the corresponding PWUW-RW-$\mathbb{N}$ and PWUW-$\mathbb{N}$ problems belong to P. We now present a lemma that for a given $\mathscr{E}$-PWUW-$\mathbb{N}$ instance $I$ and some scoring protocol $\mathscr{E}$ allows us to make a statement about which weights for which votes in $V_0$ are greater than 0.

**Lemma 3.5.** *Let $(C, V_1, V_0, c)$ be an instance of $\mathscr{E}$-PWUW-$\mathbb{N}$ for some scoring protocol $\mathscr{E}$. If there exist weights $w_i \in \mathbb{N}$ for $v_i \in V_0$, $1 \leq i \leq |V_0|$, such that $c$ wins the weighted $\mathscr{E}$ election $(C, V_1 \cup V_0)$, then there exists an alternative weight assignment, where $c$ still wins the election, with weights $w_i' \in \mathbb{N}$ for the votes in $V_0$ such that $w_i' > 0$ holds if and only if $v_i$ assigns a positive score to $c$.*

**Proof.** Let $(C, V_1 \cup V_0)$ be an election as described in the lemma and let $c$ be a winner under some weight assignment $w_i \in \mathbb{N}$ for $v_i \in V_0$, $1 \leq i \leq |V_0|$. Now, as $c$ is a winner of the election, every candidate $d \in C$ satisfies

$$score_{V_1}(c) + score_{V_0}(c) \geq score_{V_1}(d) + score_{V_0}(d). \tag{1}$$

Let $\tilde{v} \in V_0$ be a vote assigning score 0 to $c$ with weight $\tilde{w} \in \mathbb{N}$ greater than 0. Changing $\tilde{w}$ to 0 does not decrease the score of $c$ but only that of other candidates in $C \setminus \{c\}$, so (1) still holds for all candidates. Repeating this step for every vote from $V_0$ with positive weight and score 0 for $c$ yields an alternative weight assignment for the votes in $V_0$ where $c$ still wins, but only votes assigning $c$ a positive score have positive weight. ❑

## 4. Results for Nonnegative Rational Weights

In this section we study our four variants of the possible winner with uncertain weights problem for nonnegative rational weights, i.e., we study the four variants of $\mathscr{E}$-PWUW-$\mathbb{Q}^+$. We will make use of the following results.

Chamberlin and Cohen [18] observed that various voting rules can be represented by systems of linear inequalities, see also the work of Faliszewski et al. [32, 34] that uses integer linear programming to obtain fixed-parameter tractability results for certain voting problems. We make use of this result to formulate linear programs (LP) for some of the voting rules we cover in our work. Doing so enables us to solve the PWUW problem variants with rational weights for these voting rules efficiently, as long as the size of the systems describing the voting rules is polynomially bounded. To solve the resulting systems of linear inequalities efficiently, we exploit the fact that for rational instead of integer values the LINEAR PROGRAMMING problem can be solved in polynomial time [40].

However, the question is to which voting rules can we apply this technique. The crucial requirement a voting rule needs to satisfy is that the scoring function used for winner determination can be described by linear inequalities and that this description is in a certain sense independent of the votes' weights. By "independent of the votes' weights" we mean that the points a candidate gains from a vote are determined essentially in the same way in both a weighted and an unweighted electorate, but in the former we have a weighted sum of these points that gives the candidate's score, whereas in the latter we have a plain sum. The following example tries to illustrate this requirement.

**Example 4.1.** *Let $C = \{a,b,c\}$ be a set of candidates and the list of votes specified as $V = (v_1, v_2, v_3, v_4)$ with*

$$v_1 = a > c > b, \quad v_2 = c > b > a, \quad v_3 = b > a > c, \quad v_4 = c > a > b.$$

*If this is an unweighted* 2-approval *election, the score of candidate $a$ is*

$$score_V(a) = \sum_{i=1}^{4} score_{v_i}(a) = 1+0+1+1 = 3.$$

*When we add weights $w_1 = 1$, $w_2 = 3$, $w_3 = 4$, and $w_4 = 2$ to obtain a weighted election, we can calculate the score of candidate $a$ as*

$$score_V(a) = \sum_{i=1}^{4} w_i \cdot score_{v_i}(a) = 1 \cdot 1 + 3 \cdot 0 + 4 \cdot 1 + 2 \cdot 1 = 7.$$

*Now, the score of $a$ corresponds to a weighted sum of the points obtained by the unweighted votes.*

*If this would have been a* Copeland$^{1/2}$ *election, however, the score of candidate $a$ in an unweighted election can be computed as follows. For every candidate $x \in C \setminus \{a\}$ and for every vote, we check whether $a$ appears before $x$. If $a$ appears before $x$, this vote brings a point of* 1*; otherwise, a point of* $-1$*. Adding up the points over all votes*

Table 1: Score of candidate $a$ ...

(a) ... in an unweighted Copeland$^{1/2}$ election.

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ | $\sum$ | score |
|---|---|---|---|---|---|---|
| $b$ | 1 | $-1$ | $-1$ | 1 | 0 | $1/2$ |
| $c$ | 1 | $-1$ | 1 | $-1$ | 0 | $1/2$ |
| | | | | | | 1 |

(b) ... in a weighted Copeland$^{1/2}$ election.

| | $v_1$ | $v_2$ | $v_3$ | $v_4$ | $\sum$ | score |
|---|---|---|---|---|---|---|
| $b$ | 1 | $-3$ | $-4$ | 2 | $-4$ | 0 |
| $c$ | 1 | $-3$ | 4 | $-2$ | 0 | $1/2$ |
| | | | | | | $1/2$ |

*tells us if $a$ wins the pairwise comparison against $x$. If so, $a$ obtains a score of* 1*, in a tie both candidates obtain a score of* $1/2$*, and if $a$ loses, $a$ obtains a score of* 0 *and $x$ of* 1*. The sum over these scores defines* $\text{score}_V(a)$*, see Table 1a.*

*In a weighted election, we calculate $a$'s score similarly, but the votes yield points according to their weights, see Table 1b. It turns out that we have* $\text{score}_V(a) = 1$ *in the unweighted election and* $\text{score}_V(a) = 1/2$ *in the weighted election. It thus is clear that here the score of the weighted election does not correspond to a weighted sum of the scores of the unweighted election.*

Scoring functions satisfying this condition are said to be *weight-independent*. This requirement is fulfilled by, e.g., the scoring functions of all scoring rules, Bucklin, and fallback voting. Copeland's scoring function, on the other hand, does not satisfy it, as we have seen in Example 4.1. In a Copeland election, every candidate gets one point for each other candidate she beats in a pairwise contest. Who of the two candidates wins a pairwise contest and thus gains a Copeland point depends directly on the votes' weights. Thus, the Copeland score in a weighted election is not a weighted sum of the Copeland scores in the corresponding unweighted election in the above sense.

In what follows, we consider elections where the vote list consists of the two sublists $V_0$ and $V_1$. We have to assign weights $x_1, \ldots, x_{|V_0|}$ to the votes in $V_0$. We don't exclude the case where weight zero can be assigned, but we will seek to find solutions where all weights are strictly positive. For a candidate $c \in C$, let $\rho_i^0(c)$ denote the position of $c$ in the preference of the $i$-th vote in $V_0$, $1 \leq i \leq |V_0|$, and let $\rho_j^1(c)$ denote the position of $c$ in the preference of the $j$-th vote in $V_1$, $1 \leq j \leq |V_1|$. Theorem 4.2 establishes our general result that all four problem variants of PWUW-$\mathbb{Q}^+$ can be solved efficiently for voting rules with weight-independent scoring functions.

**Theorem 4.2.** *Let $\mathscr{E}$ be a voting rule with a weight-independent scoring function that can be described by a system of polynomially many linear inequalities. Then $\mathscr{E}$-*PWUW-$\mathbb{Q}^+$*, $\mathscr{E}$-*PWUW-BW-$\mathbb{Q}^+$*, $\mathscr{E}$-*PWUW-RW-$\mathbb{Q}^+$*, and $\mathscr{E}$-*PWUW-BW-RW-$\mathbb{Q}^+$ *belong to* P.

**Proof.** Let $I = (C, V_1, V_0, c, B, R)$ be an $\mathscr{E}$-PWUW-BW-RW-$\mathbb{Q}^+$ instance. Since $\mathscr{E}$ is weight-independent, there is a system of polynomially many linear inequalities $A$ such that, when satisfied, $c$ wins the $\mathscr{E}$ election $(C, V_1 \cup V_0)$. We denote by $x_1, \ldots, x_{|V_0|}$ the variables of $A$ that assign weights to the votes in $V_0$. To emphasize that $A$ uses the variables $x_1, \ldots, x_{|V_0|}$, we write $A(x_1, \ldots, x_{|V_0|})$ in the LP below. We define the *vector*

*of variables* of our LP as $\vec{x} = (x_1, \ldots, x_{|V_0|}, \chi) \in (\mathbb{Q}^+)^{|V_0|+1}$. The following LP can be used to solve $I$:

$$\begin{aligned}
&\max\ \chi, && \\
&A(x_1, \ldots, x_{|V_0|}), && (2)\\
&x_i - \chi \geq 0, && \text{for } 1 \leq i \leq |V_0|, \quad (3)\\
&\chi \geq 0, && (4)\\
&\sum_{i=1}^{|V_0|} x_i \leq B, && (5)\\
&x_i \leq r_i, && \text{for } 1 \leq i \leq |V_0|, \quad (6)\\
&-x_i \leq -\ell_i, && \text{for } 1 \leq i \leq |V_0|. \quad (7)
\end{aligned}$$

Constraint (2) lists the linear inequalities that have to be fulfilled in $A$ for the distinguished candidate $c$ to win under $\mathscr{E}$. By maximizing $\chi$ via the objective function we try to find solutions where the weights are positive; this is accomplished by constraints (3) and (4). Constraint (5) implements our given upper bound $B$ for the total weight to be assigned to votes in $V_0$ and constraints (6) and (7) implement our given ranges $R_i = [\ell_i, r_i] \subseteq \mathbb{N}$ for each weight.

When we omit constraint (5), the LP solves $\mathscr{E}$-PWUW-RW-$\mathbb{Q}^+$ instances. Omitting instead constraints (6) and (7) the LP solves $\mathscr{E}$-PWUW-BW-$\mathbb{Q}^+$ instances, and omitting constraints (5), (6), and (7) the LP solves $\mathscr{E}$-PWUW-$\mathbb{Q}^+$ instances.

As already mentioned earlier, a solution in $\mathbb{Q}^+$ for an LP with polynomially bounded many linear inequalities can be found in polynomial time, see Hačijan [40]. ❑

With this general theorem, we know that when we can represent a voting rule $\mathscr{E}$ via polynomially many linear inequalities, it follows that for this voting rule all four PWUW-$\mathbb{Q}^+$ variants can be solved in polynomial time. In the following corollaries we present the specific systems of linear inequalities describing general scoring rules, the voting systems Bucklin and fallback, and plurality with runoff. Combining these systems with Theorem 4.2 then yields that the corresponding problems can be solved efficiently in P.

**Corollary 4.3.** *For each scoring protocol $\mathscr{E}$ with scoring vector $\vec{\alpha}$, the problems $\mathscr{E}$-*PWUW-$\mathbb{Q}^+$*, $\mathscr{E}$-*PWUW-BW-$\mathbb{Q}^+$*, $\mathscr{E}$-*PWUW-RW-$\mathbb{Q}^+$*, and $\mathscr{E}$-*PWUW-BW-RW-$\mathbb{Q}^+$ *belong to* P.

**Proof.** Let $(C, V_1 \cup V_0)$ be an election with $m = |C|$ candidates and $c \in C$ as the distinguished candidate. Recall that $\rho_i^0(c)$ denotes $c$'s position in the preference of vote $v_i \in V_0$, that $\alpha_{\rho_i^0(c)}$ denotes the number of points $c$ gets for this position according to the scoring vector $\vec{\alpha}$, and that in our setting all votes in $V_1$ have weight one, although this could be generalized in a straightforward way if required. Then the distinguished candidate $c$ is a winner of the election if and only if for all candidates $c' \in C \setminus \{c\}$, we

have

$$\sum_{j=1}^{|V_0|} x_j \cdot \left(\alpha_{\rho_j^0(c)} - \alpha_{\rho_j^0(c')}\right) \geq \mathit{diff}_{(C,V_1)}(c',c),$$

where $x_j$, $1 \leq j \leq |V_0|$, is the weight that will be assigned to vote $v_j \in V_0$. Consequently, every scoring protocol $\mathscr{E}$ with scoring vector $\vec{\alpha}$ can be described by a system of $m-1$ linear inequalities for $m$ candidates, and hence satisfies the requirements of Theorem 4.2. We now provide the complete system of linear inequalities required to solve a corresponding $\mathscr{E}$-PWUW-BW-RW-$\mathbb{Q}^+$ instance. As in the proof of Theorem 4.2, we have the vector of variables $\vec{x} = (x_1, x_2, \ldots, x_{|V_0|}, \chi) \in (\mathbb{Q}^+)^{|V_0|+1}$. The LP for scoring vector $\vec{\alpha}$ is of the following form:

$$\max\ \chi,$$

$$\sum_{j=1}^{|V_0|} x_j \cdot \left(\alpha_{\rho_j^0(c')} - \alpha_{\rho_j^0(c)}\right) \leq \mathit{diff}_{(C,V_1)}(c,c'), \quad \forall c' \in C \setminus \{c\}, \tag{8}$$

$$x_j - \chi \geq 0, \qquad \text{for } 1 \leq j \leq |V_0|, \tag{9}$$

$$\chi \geq 0, \tag{10}$$

$$\sum_{j=1}^{|V_0|} x_j \leq B, \tag{11}$$

$$x_j \leq r_j, \qquad \text{for } 1 \leq j \leq |V_0|, \tag{12}$$

$$-x_j \leq -\ell_j, \qquad \text{for } 1 \leq j \leq |V_0|. \tag{13}$$

Again, constraints (11) to (13) are needed only for the restricted variants. This LP can be solved in polynomial time, since we have at most $(m-1)|V_0| + 3|V_0| + 2 = (m+2)|V_0| + 2$ constraints. ❑

Note that adding $\chi$ to the left-hand side of (8) provides a way to transform our approach to the unique-winner model such that, when $\chi$ is positive, one has a weight assignment making the distinguished candidate $c$ a unique winner for the given election.

Being level-based voting rules, for Bucklin and fallback voting we have to slightly expand the presented approach. Intuitively, it is clear that we first try to make the distinguished candidate a level 1 winner. If this attempt fails, we try the second level, and so on. For Bucklin voting, the representation by linear inequalities is due to Dorn and Schlotter [24], and we adapt it here to the simplified versions of Bucklin and fallback voting. For the latter, we add appropriate constraints if the approval stage is reached. Note that the proof of Corollary 4.4 cannot simply be adjusted to work in the unique-winner case.

**Corollary 4.4.** *Let $\mathscr{E}$ be either Bucklin or fallback voting. The problems $\mathscr{E}$-*PWUW-$\mathbb{Q}^+$*, $\mathscr{E}$-*PWUW-BW-$\mathbb{Q}^+$*, $\mathscr{E}$-*PWUW-RW-$\mathbb{Q}^+$*, and $\mathscr{E}$-*PWUW-BW-RW-$\mathbb{Q}^+$ *are all in* P.

**Proof.** We begin by first showing the result for Bucklin voting. Afterwards, we adjust the proof for fallback voting. As already said ahead of the proof, the linear inequalities for Bucklin voting are due to Dorn and Schlotter [24]. We use their approach, adjusted to our notations. Let $I = (C, V_1, V_0, c, B, R)$ be a PWUW-BW-RW-$\mathbb{Q}^+$ instance. For every possible Bucklin score $\ell$, $1 \leq \ell \leq m = |C|$, we try to solve the following LP over $\mathbb{Q}^+$:

$$\max\ \chi,$$

$$\left\lceil \frac{|V_1| + \sum_{i=1}^{|V_0|} x_i}{2} \right\rceil - 1 - \left( \sum_{i=1}^{|V_1|} \mathbb{1}_{[\rho_i^1(c') \leq \ell - 1]} + \sum_{i=1}^{|V_0|} x_i \cdot \mathbb{1}_{[\rho_i^0(c') \leq \ell - 1]} \right) \geq 0, \qquad \forall c' \in C \setminus \{c\}, \tag{14}$$

$$\left( \sum_{i=1}^{|V_1|} \mathbb{1}_{[\rho_i^1(c) \leq \ell]} + \sum_{i=1}^{|V_0|} x_i \cdot \mathbb{1}_{[\rho_i^0(c) \leq \ell]} \right) - \left\lceil \frac{|V_1| + \sum_{i=1}^{|V_0|} x_i}{2} \right\rceil \geq 0, \tag{15}$$

$$x_i - \chi \geq 0, \quad \text{for } 1 \leq i \leq |V_0|, \tag{16}$$

$$\chi \geq 0, \tag{17}$$

$$\sum_{i=1}^{|V_0|} x_i \leq B, \tag{18}$$

$$x_i \leq r_i, \quad \text{for } 1 \leq i \leq |V_0|, \tag{19}$$

$$-x_i \leq -\ell_i, \text{ for } 1 \leq i \leq |V_0|. \tag{20}$$

In the $\ell$-th LP, we look for weights $x_1, \ldots, x_{|V_0|} \in \mathbb{Q}^+$ such that $c$ has a Bucklin score of $\ell$, constraint (15), while no other candidate $c' \in C \setminus \{c\}$ has a Bucklin score smaller than $\ell$, constraint (14). Similar to the previous proofs, constraint (18) ensures that the sum of the weights allocated satisfies the upper bound $B$ and constraints (19) and (20) ensure that all weights are chosen from their corresponding ranges. If for one value $1 \leq \ell \leq m$ we obtain a feasible solution for the corresponding LP, we know that $I$ is a YES-instance. Since we search for solutions in $\mathbb{Q}^+$, it follows by the same arguments as for the previous proofs that these LPs can be solved in polynomial time. Therefore, PWUW-BW-RW-$\mathbb{Q}^+$ belongs to P for Bucklin voting. Again, removing the corresponding constraints, we see that PWUW-BW-$\mathbb{Q}^+$, PWUW-RW-$\mathbb{Q}^+$, and PWUW-$\mathbb{Q}^+$ belong to P, too.

For fallback voting, we adjust the previously introduced LP as follows. Note that nontotal votes are allowed for fallback voting; let $\ell_m$ denote the length of a longest vote present in the given instance $I$. We try to solve the same LP as for Bucklin voting, but this time only for the values $\ell$, $1 \leq \ell \leq \ell_m$. Doing so, we try to find weights $x_1, \ldots, x_{|V_0|} \in \mathbb{Q}^+$ for the votes in $V_0$ such that $c$ wins the election through her Bucklin score. If none of the LPs is feasible, we can replace constraint (15) by the following

constraint

$$\sum_{i=1}^{|V_0|} x_i \cdot \left( \mathbb{1}_{[\rho_i^0(c) \leq \ell_m]} - \mathbb{1}_{[\rho_i^0(c') \leq \ell_m]} \right) \leq \sum_{i=1}^{|V_1|} \mathbb{1}_{[\rho_i^1(c') \leq \ell_m]} - \mathbb{1}_{[\rho_i^1(c') \leq \ell_m]}, \forall c' \in C \setminus \{c\}. \tag{21}$$

Executing the modified LP with $\ell = \ell_m$, we try to make $c$ a fallback winner, i.e., $c$'s level $\ell_m$ score shall be at least as high as the level $\ell_m$ score of all other candidates, while we ensure by constraint (14) that no candidate $c' \in C \setminus \{c\}$ is accidentally made a Bucklin winner for some Bucklin score $\ell < \ell_m$. Consequently, it follows by the previous arguments that PWUW-BW-RW-$\mathbb{Q}^+$ as well as the other four variants can be solved in polynomial time for fallback voting. ❑

Finally, we provide our results for plurality with runoff and veto with runoff, taking an approach similar to the previous one: For each candidate $d$ different from $c$, we use a set of linear inequalities to figure out whether there exists a set of weights such that (1) $c$ and $d$ enter the runoff (i.e., the plurality or veto scores of $c$ and $d$ are at least as high as the scores of all other candidates), and (2) $c$ beats $d$ in their pairwise contest.

**Corollary 4.5.** *For both plurality with runoff and veto with runoff, the problems* PWUW-$\mathbb{Q}^+$, PWUW-BW-$\mathbb{Q}^+$, PWUW-RW-$\mathbb{Q}^+$, *and* PWUW-BW-RW-$\mathbb{Q}^+$ *are all in* P.

**Proof.** Let $(C, V_1 \cup V_0)$ be an election and $c \in C$ the distinguished candidate. Let PR be a shorthand for plurality with runoff and VR a shorthand for veto with runoff. We prove the results for both rules, PR and VR, at the same time as the only difference between both rules is the scoring vector $\vec{\alpha}$ used to calculate scores.

A candidate $z \in C$ enters the runoff if for all $y \in C \setminus \{z\}$, it holds that

$$\sum_{i=1}^{|V_0|} x_i \cdot (\alpha_{\rho_i^0(z)} - \alpha_{\rho_i^0(y)}) \geq \mathit{diff}_{(C,V_1)}(z,y),$$

in other words, $z$ enters the runoff if she has a score at least as high as that of all other candidates $y \in C \setminus \{z\}$. Assuming that $z$ and $y$ enter the runoff, $z$ wins the runoff if

$$\sum_{i=1}^{|V_0|} x_i \cdot (\alpha_{\kappa_i^0(z)} - \alpha_{\kappa_i^0(y)}) \geq \mathit{diff}_{(\{z,y\},V_1)}(z,y),$$

where $\kappa_i^0(c) \in \{1,2\}$ describes the position of candidate $c$ in vote $v_i \in V_0$ when reduced to the two candidates in the runoff. Consequently, the following LP checks whether there are nonnegative weights $x_1, \ldots, x_{|V_0|} \in \mathbb{Q}^+$ for the votes in $V_0$ such that $c$ and $d$

enter the runoff and $c$ wins the runoff:

$$\max \chi,$$

$$\sum_{i=1}^{|V_0|} x_i \cdot (\alpha_{\rho_i^0(c)} - \alpha_{\rho_i^0(z)}) \geq \mathit{diff}_{(C,V_1)}(c,z), \qquad \forall z \in C \setminus \{c,d\}, \tag{22}$$

$$\sum_{i=1}^{|V_0|} x_i \cdot (\alpha_{\rho_i^0(d)} - \alpha_{\rho_i^0(z)}) \geq \mathit{diff}_{(C,V_1)}(d,z), \qquad \forall z \in C \setminus \{c,d\}, \tag{23}$$

$$\sum_{i=1}^{|V_0|} x_i \cdot (\alpha_{\kappa_i^0(c)} - \alpha_{\kappa_i^0(d)}) \geq \mathit{diff}_{(\{c,d\},V_1)}(c,d), \tag{24}$$

$$x_i - \chi \geq 0, \qquad \text{for } 1 \leq i \leq |V_0|, \tag{25}$$

$$\chi \geq 0, \tag{26}$$

$$\sum_{i=1}^{|V_0|} x_i \leq B, \tag{27}$$

$$x_i \leq r_i, \qquad \text{for } 1 \leq i \leq |V_0|, \tag{28}$$

$$-x_i \leq -\ell_i, \qquad \text{for } 1 \leq i \leq |V_0|. \tag{29}$$

This LP has at most $2(|C|-2)+3|V_0|+3$ inequalities; hence, its size is polynomial in the size of the election. Consequently, we can execute this LP efficiently for every candidate $d \in C \setminus \{c\}$ to check whether there exists a candidate that can make it into the runoff together with $c$ such that $c$ wins against this candidate in the runoff. If such a candidate exists, we have a YES- and otherwise a NO-instance of our problem. Obviously, with this approach we can efficiently solve PR- and VR-PWUW-BW-RW-$\mathbb{Q}^+$ instances. Omitting constraint (27), we can solve PWUW-RW-$\mathbb{Q}^+$ instances. Omitting constraints (28) and (29), we can solve PWUW-BW-$\mathbb{Q}^+$ instances. When we omit all three constraints, we solve PWUW-$\mathbb{Q}^+$ instances. Thus all four problem variants can be solved efficiently for both voting rules and, hence, belong to P. ❑

Note that the proof of Corollary 4.5 does not work in the unique-winner model. Obviously, we can extend this proof to any scoring-vector-based voting rule with runoff. However, since besides plurality and veto there are no other commonly used voting rules with runoff, we did not generalize Corollary 4.5 and kept it specific to these two rules.

With this proof we conclude our section related to results for nonnegative rational weights. To summarize, we have proven a general theorem yielding tractability results for all four variants of $\mathscr{E}$-PWUW-$\mathbb{Q}^+$ for voting rules $\mathscr{E}$ whose scoring functions satisfy weight independence. Afterwards, we have applied this theorem to four central voting rules, showing sixteen P membership results in total. Table 7 in Section 6 provides an overview of all our results related to nonnegative rational weights.

## 5. Results for Nonnegative Integer Weights

Having presented all our results for the possible winner with uncertain weights problem for nonnegative rational weights in the previous section, in this section we turn

to nonnegative integer weights. Each of the next subsections will present our results for another voting rule. Chronologically, we present our results for $k$-approval, $k$-veto, plurality with runoff, veto with runoff, Copeland$^\alpha$, ranked pairs, Bucklin voting, fallback voting, and Borda. For each of these voting rules, we provide computational complexity results for all four variants of the PWUW-$\mathbb{N}$ problem.

Before we begin to study specific voting rules, we prove Lemma 5.1 as an auxiliary means, which holds for all scoring protocols $\mathscr{E}$ and provides a way of preprocessing our PWUW-BW-RW-$\mathbb{N}$ problem instances.

**Lemma 5.1.** *Let $I = (C, V_1, V_0, c, B, R)$ be an $\mathscr{E}$-PWUW-BW-RW-$\mathbb{N}$ instance for a scoring protocol $\mathscr{E}$. Then there exists an instance $I' = (C, V_1', V_0, c, B', R')$ with $l_i' = 0$ for all $R_i' = [l_i', r_i'] \in R'$ such that $I \in \mathscr{E}$-PWUW-BW-RW-$\mathbb{N}$ if and only if $I' \in \mathscr{E}$-PWUW-BW-RW-$\mathbb{N}$.*

**Proof.** Let $I = (C, V_1, V_0, c, B, R)$ be an $\mathscr{E}$-PWUW-BW-RW-$\mathbb{N}$ instance. We construct the desired instance $I' = (C, V_1', V_0, c, B', R')$ as follows. For every $R_i = [l_i, r_i] \in R$, we define $R_i' = [0, r_i']$ with $r_i' = r_i - l_i$ and add it to $R'$. Further, we add $l_i$ copies of the corresponding vote $v_i \in V_0$ with weight 1 to $V_1'$ and define $B' = B - \sum_{i=1}^{|V_0|} l_i$.

Now, assume that $I$ is a YES-instance of $\mathscr{E}$-PWUW-BW-RW-$\mathbb{N}$. Then there exist weights $w_i \in \mathbb{N}$ for $1 \leq i \leq |V_0|$ such that $w_i \in R_i$ and $\sum_{i=1}^{|V_0|} w_i \leq B$ and $c$ wins the weighted election $(C, V_1 \cup V_0)$. For $1 \leq i \leq |V_0|$, we define the weights $w_i' = w_i - l_i \geq 0$. Consequently, we have

$$w_i' = w_i - l_i \in R_i - l_i = [l_i - l_i, r_i - l_i] = [0, r_i'] = R_i'$$

as well as

$$\sum_{i=1}^{|V_0|} w_i' = \sum_{i=1}^{|V_0|} (w_i - l_i) = \sum_{i=1}^{|V_0|} w_i - \sum_{i=1}^{|V_0|} l_i \leq B - \sum_{i=1}^{|V_0|} l_i = B'.$$

Let $v_i$ be a vote from $V_0$ with weight $w_i$. Then this vote is added with a weight of $w_i$ to the election in $I$. Now, with the altered weights in $I'$, the vote is added with a weight of $w_i' = w_i - l_i$ to the election in $I'$. However, as previously described, we added $l_i$ copies of $v_i$ with weight 1 to $V_1'$. In total, $v_i$ together with its copies in $V_1'$ thus obtains a weight of $w_i' + l_i = w_i$. Hence, the votes' overall weight has not changed from $I$ to $I'$ and, thus, $c$ wins the weighted election $(C, V_1' \cup V_0)$ and $I'$ is a YES-instance of $\mathscr{E}$-PWUW-BW-RW-$\mathbb{N}$ as well.

Conversely, let us assume that $I'$ is a YES-instance of $\mathscr{E}$-PWUW-BW-RW-$\mathbb{N}$. Then there exist weights $w_i' \in \mathbb{N}$ for $1 \leq i \leq |V_0|$ such that $c$ wins the weighted election $(C, V_1' \cup V_0)$. For every weight $w_i'$, we define the weight $w_i = w_i' + l_i$. Afterwards, we have

$$w_i = w_i' + l_i \in R_i' + l_i = [0 + l_i, r_i' + l_i] = [l_i, r_i] = R_i$$

as well as

$$\sum_{i=1}^{|V_0|} w_i = \sum_{i=1}^{|V_0|} (w_i' + l_i) = \sum_{i=1}^{|V_0|} w_i' + \sum_{i=1}^{|V_0|} l_i \leq B' + \sum_{i=1}^{|V_0|} l_i = B.$$

Following the same argument as before, the candidates' overall scores do not change and, therefore, candidate $c$ wins the weighted election $(C, V_1 \cup V_0)$, so that $I$ is a YES-instance of $\mathscr{E}$-PWUW-BW-RW-$\mathbb{N}$, too. ❑

We now provide an argument that we will use in some upcoming proofs related to PWUW-BW-RW-$\mathbb{N}$ so as to argue why certain assumptions can be made.

**Remark 5.2.** *Let $\mathscr{E}$ be a binary scoring protocol and $I = (C, V_1, V_0, c, B, R)$ an instance for $\mathscr{E}$-PWUW-BW-RW-$\mathbb{N}$. Furthermore, let us denote by $V_0' \subseteq V_0$ the subset of votes from $V_0$ that assign a positive score to $c$, and let $R_i = [l_i, r_i] \in R$ denote the region of $v_i \in V_0'$. Now, if it holds that*

$$\sum_{i=1}^{|V_0'|} r_i \leq B,$$

*it immediately follows that $I$ is a trivial instance. Following the argument in the proof of Lemma 3.5, it only makes sense to allocate a positive weight to votes in $V_0'$ if one wants $c$ to win the weighted election $(C, V_1 \cup V_0)$. However, if the sum of the upper limits of these votes is smaller than $B$, we can assign all these votes their maximal weights and all other votes in $V_0 \setminus V_0'$ a weight of 0. Either $c$ wins this election, i.e., $I$ would be a* YES*-instance of $\mathscr{E}$-*PWUW-BW-RW-$\mathbb{N}$*, or loses it, i.e., $I$ would be a* NO*-instance of $\mathscr{E}$-*PWUW-BW-RW-$\mathbb{N}$*. Consequently, to exclude trivial instances, we can always assume that the sum of the upper limits of the votes in $V_0'$ is greater than $B$.*

*Additionally, excluding trivial instances, if we know that instance $I$ is a* YES*-instance, we can also assume that $\sum_{i=1}^{|V_0'|} w_i = B$ holds, as $\sum_{i=1}^{|V_0'|} r_i \geq B$ is true, so that we can always increase some of the weights $w_i$ until the sum of these weights equals $B$ without altering the outcome of the election, as we only increase the weights of votes adding positive scores to $c$.*

### *5.1. k-Approval*

We start by studying the voting rule $k$-approval. Recall that $k$-approval is a scoring protocol with scoring vector

$$\vec{\alpha} = (\underbrace{1, \ldots, 1}_{k}, 0, \ldots, 0),$$

where $\vec{\alpha}$ has as many entries as there are candidates in an election, with the first $k$ entries being a 1 and the remaining entries being a 0. Obviously, $k$-approval is also a binary scoring protocol, as defined in Section 2, so Theorem 3.4 applies. Table 2 provides an overview of how the results on $k$-approval are structured.

From Theorem 3.4 we immediately obtain Corollary 5.3, which already settles the complexity of $k$-approval-PWUW-$\mathbb{N}$ and $k$-approval-PWUW-RW-$\mathbb{N}$ for all $k \geq 1$.

**Corollary 5.3.** *For every $k \geq 1$, the problems* $k$-approval-PWUW-$\mathbb{N}$ *and* $k$-approval-PWUW-RW-$\mathbb{N}$ *belong to* P.

Table 2: Overview of results proving P membership for our four PWUW variants of $k$-approval.

| $k$-approval-PWUW | $k=1$ | $k=2$ | $k=3$ | $k\geq 4$ |
|---|---|---|---|---|
| -$\mathbb{N}$ | Cor. 5.3 | Cor. 5.3 | Cor. 5.3 | Cor. 5.3 |
| -RW-$\mathbb{N}$ | Cor. 5.3 | Cor. 5.3 | Cor. 5.3 | Cor. 5.3 |
| -BW-$\mathbb{N}$ | Prop. 5.4 | Prop. 5.5 | Cor. 5.7 | Thm. 5.8 |
| -BW-RW-$\mathbb{N}$ | Prop. 5.4 | Prop. 5.5 | Thm. 5.6 | Thm. 5.8 |

It remains to study the complexity of $k$-approval-PWUW-BW-$\mathbb{N}$ and $k$-approval-PWUW-BW-RW-$\mathbb{N}$; we will first consider small values of $k$ and then proceed to larger values of $k$. Starting with $k=1$, the next proposition provides results for the two remaining cases of 1-approval, also known as plurality, not resolved by the previous corollary, namely plurality-PWUW-BW-$\mathbb{N}$ and plurality-PWUW-BW-RW-$\mathbb{N}$.

**Proposition 5.4.** 1-approval-PWUW-BW-$\mathbb{N}$ *and* 1-approval-PWUW-BW-RW-$\mathbb{N}$ *belong to* P.

**Proof.** First, we prove the result for 1-approval-PWUW-BW-$\mathbb{N}$ and at the end of this proof we extend it to also hold for 1-approval-PWUW-BW-RW-$\mathbb{N}$. Let $I=(C,V_1,V_0,c,B)$ be a 1-approval-PWUW-BW-$\mathbb{N}$ instance. To determine if there are weights for the votes in $V_0$ such that $c$ can win the weighted election, we proceed as follows. For every candidate $d\in C\setminus\{c\}$, we calculate the difference of scores between $d$ and $c$ in $V_1$, i.e.,

$$\mathit{diff}_{(C,V_1)}(d,c)=\mathit{score}_{V_1}(d)-\mathit{score}_{V_1}(c).$$

We denote by

$$\rho=\max_{d\in C\setminus\{c\}}\mathit{diff}_{(C,V_1)}(d,c)$$

a greatest difference between the scores of some candidate $d$ in $C\setminus\{c\}$ and $c$. If $\rho\leq 0$, $c$ is already among the winners of the unweighted election and we obtain a YES-instance of 1-approval-PWUW-BW-$\mathbb{N}$ by setting the weights of all votes in $V_0$ to 0. Otherwise (i.e., if $\rho>0$), we need to validate whether there is at least one vote $v\in V_0$ with $c$ on its first position as well as whether $\rho\leq B$. If both conditions are satisfied, $I$ is a YES-instance of 1-approval-PWUW-BW-$\mathbb{N}$, as we can simply assign a weight of $\rho$ to $v$, so that $c$ is among the winners of the weighted election $(C,V_1\cup V_0)$. That is so because $c$ gains a score of $\rho$ from this weight assignment, while all other candidates' scores stay constant. Otherwise (i.e., if there is no vote $v\in V_0$ with $c$ on its first position or $\rho>B$ is true), $I$ is a NO-instance of 1-approval-PWUW-BW-$\mathbb{N}$: If there is no vote $v\in V_0$ with $c$ on top, it is impossible to increase $c$'s score via votes from $V_0$; if $\rho>B$, then $c$ cannot win the weighted election as there is at least one candidate who has a point advantage over $c$ from $V_1$ greater than $B$. But as we can increase $c$'s score via votes from $V_0$ at most by $B$, $c$ can never reach a high enough score to be among the weighted election's winners. All the previously described calculations can be executed in time polynomial in $|I|$, so it follows that 1-approval-PWUW-BW-$\mathbb{N}$ is in P.

Now, we extend the proof to also work for 1-approval-PWUW-BW-RW-$\mathbb{N}$. Let $I=(C,V_1,V_0,c,B,R)$ be a 1-approval-PWUW-BW-RW-$\mathbb{N}$ instance; by Lemma 5.1, we

may assume that for every $R_i = [l_i, r_i] \in R$, we have $l_i = 0$. We proceed very similarly to the previous approach. First, we calculate $\rho = \max_{d \in C \setminus \{c\}} \mathit{diff}_{(C,V_1)}(d,c)$. Again, if $\rho \leq 0$, we know that $I$ is a YES-instance of 1-approval-PWUW-BW-RW-$\mathbb{N}$, and if $\rho > B$, we know that $I$ is a NO-instance of 1-approval-PWUW-BW-RW-$\mathbb{N}$. Finally, in the case of $0 < \rho \leq B$ we argue as follows for $I$ to be a YES-instance of 1-approval-PWUW-BW-RW-$\mathbb{N}$. Let $V_0' \subseteq V_0$ denote the subset of votes from $V_0$ with $c$ on their first position. From Corollary 5.3 we know that 1-approval-PWUW-RW-$\mathbb{N}$ belongs to P. Therefore, following the argument of Remark 5.2, we can assume that $\sum_{i=1}^{|V_0'|} r_i \geq B$. Thus, similarly as for 1-approval-PWUW-BW-$\mathbb{N}$, we assign a total weight of $\rho$ to votes from $V_0'$, so $c$ gains an additional score of $\rho$ in the weighted election $(C, V_1 \cup V_0)$ and hence, is among the winners. Again, all calculations can be executed in time polynomial in $|I|$, so it follows that 1-approval-PWUW-BW-RW-$\mathbb{N}$ is in P. ❑

Note that Bartholdi, Tovey, and Trick [2] proved that plurality-CCAV$_{\text{succ}}$ belongs to P. Consequently, using Proposition 3.3, we obtain an alternative proof that plurality-PWUW-BW-RW-$\mathbb{N}$ belongs to P. Using Theorem 3.1 we then can argue that plurality-PWUW-BW-$\mathbb{N}$ belongs to P, too.

With the complexity for all four problem variants of 1-approval-PWUW resolved, in Proposition 5.5 below we study the complexity of the two open cases for 2-approval-PWUW, namely 2-approval-PWUW-BW-$\mathbb{N}$ and 2-approval-PWUW-BW-RW-$\mathbb{N}$. To this end, we will use the following decision problem:

| MAXIMUMFLOW (MAXFLOW) | |
|---|---|
| **Given:** | A directed graph $G = (V, E)$, two distinguished vertices $s, t \in V$, an edge capacity function $\kappa\colon E \to \mathbb{N}$, and some integer $k \in \mathbb{N}$. |
| **Question:** | Is there a flow from $s$ to $t$ through $G$ of volume at least $k$? |

The algorithm by Fulkerson and Dantzig [36] provides an efficient approach to solving MAXFLOW instances, i.e., MAXFLOW $\in$ P is a well-known result. Having recalled this decision problem as well as its P membership, we can now state our results.

**Proposition 5.5.** 2-approval-PWUW-BW-$\mathbb{N}$ *as well as* 2-approval-PWUW-BW-RW-$\mathbb{N}$ *belong to* P.

**Proof.** First, we prove the result for 2-approval-PWUW-BW-RW-$\mathbb{N}$. Then we will explain how to derive the result for 2-approval-PWUW-BW-$\mathbb{N}$ from the former one.

To prove that 2-approval-PWUW-BW-RW-$\mathbb{N}$ belongs to P, we reduce this problem to MAXFLOW. Let $I = (C, V_1, V_0, c, B, R)$ be a 2-approval-PWUW-BW-RW-$\mathbb{N}$ instance. We denote by $V_0' \subseteq V_0$ the subset of votes from $V_0$ which rank $c$ among the top two positions. For each $v_i \in V_0'$, we denote by $d_i$ the candidate distinct from $c$ that is among the top two positions of $v_i$ together with $c$. With Lemma 3.5 we know that in order to check whether $c$ can win a weighted election, it is sufficient to only consider weight allocations assigning positive weight to votes in $V_0'$. By Lemma 5.1, we can assume that for all $R_i \in R$ it holds that $R_i = [0, r_i]$ and, lastly, with Remark 5.2 we can assume that $\sum_{i=1}^{|V_0'|} r_i \geq B$ holds. Having recalled all these assumptions, we are ready to construct

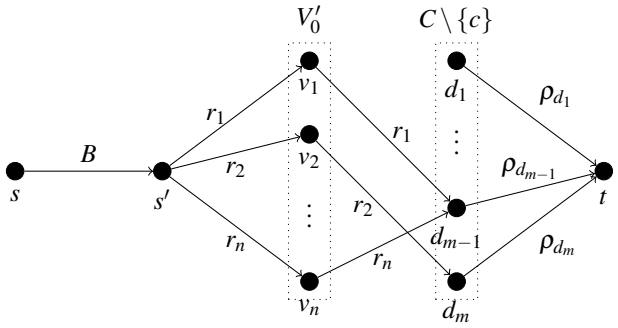



Figure 1: Illustration of a flow network created to show 2-approval-PWUW-BW-RW-$\mathbb{N} \leq_m^p$ MAXFLOW for $n$ votes in $V_0'$ and $m+1$ candidates in $C$; the edges in the network are labeled with their capacities.

a MAXFLOW instance $I' = (G, s, t, \kappa, k)$ as follows. For the directed graph $G = (V, E)$, we define the set of vertices as

$$V = \{s, s', t\} \cup V_0' \cup (C \setminus \{c\}),$$

where $s \in V$ is the source and $t \in V$ the target of the flow. Next, we add the following edges with their corresponding capacities via $\kappa$ to $E$:

1. $(s, s')$ with capacity $B$;
2. for every $v_i \in V_0'$ with $R_i = [0, r_i]$, we add $(s', v_i)$ with capacity $r_i$;
3. for every $v_i \in V_0'$, we add $(v_i, d_i)$ with capacity $r_i$; and
4. for every candidate $d \in C \setminus \{c\}$, we add an edge $(d, t)$ with capacity $\rho_d = B + \mathit{diff}_{(C,V_1)}(c, d)$.

Setting $k = B$ completes the construction of $I'$ in time polynomial in $|I|$. Figure 1 illustrates how such a constructed network $G$ might look like. We now show that $I$ is a YES-instance of 2-approval-PWUW-BW-RW-$\mathbb{N}$ if and only if $I'$ is a YES-instance of MAXFLOW.

Assume that $I$ is a YES-instance of 2-approval-PWUW-BW-RW-$\mathbb{N}$. Then there exist weights $w_i \in \mathbb{N}$ for all votes in $V_0$ such that $c$ wins the weighted election $(C, V_1 \cup V_0)$. With Lemma 3.5 we can assume a weight of 0 for all votes in $V_0 \setminus V_0'$, and because of Remark 5.2 we can furthermore assume that $\sum_{i=1}^{|V_0'|} w_i = B$. To prove that $I'$ is a YES-instance of MAXFLOW, we need to give a flow through $G$ of volume at least $B$. Obviously, as $\kappa((s, s')) = B$, we can assign a flow of volume $B$ to $(s, s')$. Next, for every $v_i \in V_0'$, we assign a flow of volume $w_i$ to $(s', v_i)$. That is possible because $w_i \leq r_i = \kappa((s', v_i))$. Furthermore, since we earlier assumed that the sum of all weights $w_i$ equals $B$, the inflow of volume $B$ to $s'$ equals its outflow to all nodes in $V_0'$ via the edges $(s', v_i)$. Next, every vertex $v_i$ (recall that $v_i$ is a vote in $V_0'$) has exactly one outgoing edge to a neighbor, namely $d_i$, the candidate which together with $c$ is among the top two positions of $v_i$. By construction, these edges $(v_i, d_i)$ have a capacity of $\kappa((v_i, d_i)) = r_i$, so every node $v_i$ can pass its total inflow volume $w_i \leq r_i$ on to $d_i$ via its single outgoing edge. Now, every node $d \in C \setminus \{c\}$ has a nonnegative inflow via the votes in $V_0'$ and

their weights $w_i$. Obviously, the inflow volume of a node $d \in C \setminus \{c\}$ corresponds to the score candidate $d$ obtains from the weighted votes in $V_0$, i.e., $score_{V_0}(d)$. Every node $d \in C \setminus \{c\}$ has exactly one outgoing edge $(d,t)$ with capacity $\rho_d$. Hence, to satisfy this capacity limit, it must hold that

$$\begin{aligned} & score_{V_0}(d) \leq \rho_d = B + diff_{(C,V_1)}(c,d) \\ \Leftrightarrow \quad & score_{V_0}(d) \leq B + score_{V_1}(c) - score_{V_1}(d) \\ \Leftrightarrow \quad & score_{V_1}(d) + score_{V_0}(d) \leq score_{V_1}(c) + B \\ \Leftrightarrow \quad & score_{V_1}(d) + score_{V_0}(d) \leq score_{V_1}(c) + score_{V_0}(c). \end{aligned} \tag{30}$$

However, as we have assumed that $I$ is a YES-instance of 2-approval-PWUW-BW-RW-$\mathbb{N}$, Equation (30) must be true for every candidate $d \in C \setminus \{c\}$, as $c$ is a winner of the weighted election $(C, V_1 \cup V_0)$. Consequently, $t$ has an inflow that corresponds to the sum of all weights $w_i$ for $v_i \in V_0'$, which is equal to $B$, and we have found a flow through $G$ of volume $k = B$, so $I'$ is a YES-instance of MAXFLOW.

Conversely, assume that $I'$ is a YES-instance of MAXFLOW. Then there exists a flow of volume $\omega \geq k = B$ through $G$. Since $s$ is the source of the flow, the flow of volume $\omega$ must start in $s$. By the capacity of $s$'s only outgoing edge $(s,s')$, namely $\kappa((s,s')) = B$, it directly follows that $\omega = B$ must hold. Now, for every node $v_i \in V_0'$, we interpret its inflow volume $\omega_i$, coming from $s'$ and satisfying $\omega_i \leq r_i = \kappa((s',v_i))$, as the weight $w_i$ of vote $v_i$ in the weighted election $(C, V_1 \cup V_0)$. Since $B = \sum_{i=1}^{|V_0'|} \omega_i$ must hold to ensure that the inflow volume of $s'$ equals its outflow volume, we know from interpreting the flows $\omega_i$ as weights $w_i$ of the votes in $V_0'$ that candidate $c$ obtains a score of $score_{V_0}(c) = B$ from the votes in $V_0'$. As the flow of volume $B$ through $G$ is valid, it follows for every node $d \in C \setminus \{c\}$ that its outflow, which then corresponds to $score_{V_0}(d)$ in the weighted election, satisfies the capacity limit, i.e., $score_{V_0}(d) \leq \rho_d$. By the same argument as for Equation (30) it follows that $c$ wins the weighted election $(C, V_1 \cup V_0)$. Furthermore, we set the weights of all votes in $V_0 \setminus V_0'$ to 0. Consequently, all region requirements from $R$ are satisfied and $B$ is the upper bound on the total sum of weights, so $I$ is a YES-instance of 2-approval-PWUW-BW-RW-$\mathbb{N}$.

Hence, 2-approval-PWUW-BW-RW-$\mathbb{N} \leq_m^p$ MAXFLOW. Since MAXFLOW is in P, 2-approval-PWUW-BW-RW-$\mathbb{N}$ is in P as well. Now, it directly follows from Theorem 3.1 that 2-approval-PWUW-BW-$\mathbb{N}$ belongs to P, too. ❑

Summing up our results so far, we now know that all four problem variants (i.e., PWUW-$\mathbb{N}$, PWUW-BW-$\mathbb{N}$, PWUW-RW-$\mathbb{N}$, and PWUW-BW-RW-$\mathbb{N}$) can be solved in polynomial time for $k$-approval with $k \in \{1,2\}$. We also know that the two problem variants $k$-approval-PWUW-$\mathbb{N}$ and $k$-approval-PWUW-RW-$\mathbb{N}$ are in P for all $k \geq 1$.

Next, we turn to the question of how hard it is to solve the other two problem variants, PWUW-BW-$\mathbb{N}$ and PWUW-BW-RW-$\mathbb{N}$, for 3-approval. We will show in Theorem 5.6 and Corollary 5.7 that they are in P, too. To this end, we will provide a reduction from PWUW-BW-RW-$\mathbb{N}$ to the problem GENERALIZED-WEIGHTED-B-EDGE-MATCHING (GWBEM), which belongs to P and is defined as follows:[4]

[4]Formally, Gabow [37] and Grötschel et al. [39, p. 259] define a maximization variant of GWBEM and

| GENERALIZED-WEIGHTED-B-EDGE-MATCHING (GWBEM) | |
|---|---|
| **Given:** | An undirected multigraph $G = (N, E)$ without loops, capacity-bounding functions $a_\ell, a_u \colon E \to \mathbb{N}$ and $b_\ell, b_u \colon N \to \mathbb{N}$, a weight function $w \colon E \to \mathbb{N}$, and a target integer $r \in \mathbb{N}$. |
| **Question:** | Does there exist a function $x \colon E \to \mathbb{N}$ with $\sum_{e \in E} w(e)x(e) \geq r$ such that for every edge $e \in E$ it holds that $a_\ell(e) \leq x(e) \leq a_u(e)$ and for every node $z \in N$ it holds that $b_\ell(z) \leq \sum_{e \in \delta(z)} x(e) \leq b_u(z)$, where $\delta(z)$ is the set of edges incident to node $z$? |

Before stating and proving that 3-approval-PWUW-BW-RW-$\mathbb{N}$ belongs to P, recall the following two auxiliary lemmas, which we will use in the proof of Theorem 5.6. First, Lemma 5.1 enables us to assume for every scoring protocol $\mathscr{E}$ that, without loss of generality, all intervals $R_i = [l_i, r_i] \in R$ fulfill $l_i = 0$ for every instance $I = (C, V_1, V_0, c, B, R)$ of $\mathscr{E}$-PWUW-BW-RW-$\mathbb{N}$. Furthermore, by Lemma 3.5 we may assume that, without loss of generality, for any 3-approval-PWUW-BW-RW-$\mathbb{N}$ instance $I = (C, V_1, V_0, c, B, R)$ where $c$ wins, only votes in $V_0$ with $c$ among the top three candidates have positive weight. With these assumptions in mind, we are now ready to prove that 3-approval-PWUW-BW-RW-$\mathbb{N}$ is efficiently solvable.

**Theorem 5.6.** 3-approval-PWUW-BW-RW-$\mathbb{N}$ *is in* P.

**Proof.** We show that 3-approval-PWUW-BW-RW-$\mathbb{N} \leq_{\mathrm{m}}^{\mathrm{p}}$ GWBEM. Let $I = (C, V_1, V_0, c, B, R)$ be a 3-approval-PWUW-BW-RW-$\mathbb{N}$ instance. By Lemma 5.1, we assume for all $v_i \in V_0$ that $R_i = [0, r_i]$ holds. We can also assume $\sum_{i=1}^{|V_0|} r_i \geq B$, since the sum over the $r_i$ provides an upper bound for the overall weight distributed among the votes in $V_0$.[5] We construct a GWBEM instance $I' = (G, a_\ell, a_u, b_\ell, b_u, w, r)$ with multigraph $G = (C', E)$ whose set of nodes $C' = C \setminus \{c\}$ consists of all candidates except $c$. Furthermore, denote the subset of votes from $V_0$ where candidate $c$ is among the top three positions by

$$V_0' = \{x_1 > x_2 > x_3 > \cdots \in V_0 \mid c \in \{x_1, x_2, x_3\}\}$$

and define the set of edges in $G$ by

$$E = \{\{x_1, x_2, x_3\} \setminus \{c\} \mid x_1 > x_2 > x_3 > \cdots \in V_0'\}.$$

That is, for every vote from $V_0$ with $c$ among the top three positions, we add an edge between the vote's remaining two candidates in the top three positions. This can result in a multigraph, of course, but is in line with the problem definition. We set the target integer $r = B$ and, for every edge $e \in E$ linked to $v_i \in V_0$, we define the edge capacity bounds by $a_\ell(e) = 0$ and $a_u(e) = r_i$ and the weight function by $w(e) = 1$ for the corresponding interval $R_i = [0, r_i]$. For every $c' \in C'$, we define the node capacity bounds

show its polynomial-time solvability, which immediately implies that GWBEM is in P. The same problem was first used by Lin [49] in the context of voting and later on also, for example, by Erdélyi et al. [30].

[5] This assumption does make sense indeed: Otherwise, the parameter $B$ would be meaningless and our instance becomes a PWUW-RW-$\mathbb{N}$ instance which, as shown in Corollary 5.3, is efficiently solvable.

by

$$b_\ell(c') = 0 \quad \text{and} \quad b_u(c') = score_{V_1}(c) + B - score_{V_1}(c').$$

We can assume for every candidate $c' \in C'$ (which is, recall, a node in $G$) that $b_u(c') \geq 0$ holds, as otherwise $I$ would be a trivial NO-instance. That is the case, since $b_u(c') < 0$ implies $score_{V_1}(c') - score_{V_1}(c) > B$, which makes it impossible for $c$ to beat $c'$ with votes from $V_0$ having a total weight of at most $B$. Obviously, the construction of $I'$ can be realized in time polynomial in $|I|$.

We prove that $I \in$ 3-approval-PWUW-BW-RW-$\mathbb{N}$ if and only if $I' \in$ GWBEM.

From left to right, assume that $I$ is a YES-instance of 3-approval-PWUW-BW-RW-$\mathbb{N}$. Consequently, there exist weights $w_i \in \mathbb{N}$ for $v_i \in V_0$, $1 \leq i \leq |V_0|$, with $w_i \in R_i = [0, r_i]$ such that $c$ wins the weighted election $(C, V_1 \cup V_0)$. Without loss of generality, we can assume that $\sum_{i=1}^{|V_0|} w_i = B$, as argued in Remark 5.2. Furthermore, by Lemma 3.5 we can assume that for every vote $v_i \in V_0$, we have $w_i > 0$ if and only if there is an edge $e_i \in E$ in the graph $G$ of our instance $I'$, i.e., if $c$ is among the top three candidates in $v_i$. We set $x(e_i) = w_i$ and obtain

$$\sum_{e_i \in E} w(e_i)x(e_i) = \sum_{e_i \in E} x(e_i) = \sum_{e_i \in E} w_i = B = r.$$

Additionally, $0 \leq w_i \leq r_i$ is true for all $i$, $1 \leq i \leq |V_0|$, and, therefore, we have

$$0 = a_\ell(e_i) \leq x(e_i) = w_i \leq a_u(e_i) = r_i$$

for every edge $e_i \in E$. Since $c$ is a winner of the weighted election $(C, V_1 \cup V_0)$, the score of $c$ over $V_1$ and $V_0$ is at least as high as the score of every other candidate $d \in C'$. Because the sum of the weights over $V_0$ equals $B$, according to Lemma 3.5 we know that the score of $c$ is equal to $score_{V_1}(c) + B$. Thus, for every candidate $d \in C'$, it must hold that

$$\begin{aligned} & score_{V_1}(d) + score_{V_0}(d) \leq score_{V_1}(c) + score_{V_0}(c) \\ \Leftrightarrow \quad & score_{V_1}(d) + score_{V_0}(d) \leq score_{V_1}(c) + B \\ \Leftrightarrow \quad & \sum_{e \in \delta(d)} x(e) \leq b_u(d), \end{aligned}$$

where $score_{V_0}(d) = \sum_{e \in \delta(d)} x(e)$, as all edges incident to $d$ correspond to the votes with positive weight in which $d$ occurs among the top three positions. Hence, we have

$$0 = b_\ell(d) \leq \sum_{e \in \delta(d)} x(e) \leq b_u(d),$$

so $I' \in$ GWBEM.

From right to left, assume that $I' \in$ GWBEM. Then there exists a function $x\colon E \to \mathbb{N}$ such that

(i) $\sum_{e \in E} x(e) \geq r = B$;
(ii) for every edge $e \in E$, it holds that $a_\ell(e) \leq x(e) \leq a_u(e)$; and

(iii) for every node $c' \in C'$, it holds that $b_\ell(c') \leq \sum_{e \in \delta(c')} x(e) \leq b_u(c')$.

For every edge $e_i \in E$, we set $w_i = x(e_i)$ for the corresponding vote $v_i \in V_0'$. The weight of all remaining votes in $V_0 \setminus V_0'$ is set to 0. Consequently, for every vote, $0 \leq w_i \leq r_i$ is satisfied. For every candidate $c' \in C'$, it holds with (iii) that

$$\begin{aligned} & \sum_{e_i \in \delta(c')} x(e_i) \leq b_u(c') \\ \Leftrightarrow \quad & \sum_{e_i \in \delta(c')} x(e_i) \leq score_{V_1}(c) + B - score_{V_1}(c') \\ \Leftrightarrow \quad & \sum_{e_i \in \delta(c')} w_i \leq score_{V_1}(c) + B - score_{V_1}(c'). \end{aligned}$$

The last inequality is equivalent to

$$score_{V_1}(c') + score_{V_0}(c') \leq score_{V_1}(c) + B. \tag{31}$$

Consequently, $c$ is a winner of the weighted election $(C, V_1 \cup V_0)$, since from (i) it follows that $score_{V_1}(c) + B \leq score_{V_1}(c) + score_{V_0}(c)$. Now, define

$$\rho = \sum_{e \in E} x(e) - B \geq 0.$$

For every candidate $c' \in C'$ with (31), it holds that

$$\begin{aligned} & score_{V_1}(c) + score_{V_0}(c) - (score_{V_1}(c') + score_{V_0}(c')) \\ \geq\ & score_{V_1}(c) + score_{V_0}(c) - (score_{V_1}(c) + B) \\ =\ & score_{V_0}(c) - B \\ =\ & \sum_{e \in E} x(e) - B = \rho. \end{aligned}$$

Hence, every candidate in $C'$ has at least $\rho$ points less than candidate $c$. Consequently, we can remove a total weight of $\rho$ from those edges in $G$ with $x(e) > 0$ (which is possible because $b_\ell(e) = 0$ for all edges $e \in E$), so we have $\sum_{e \in E} x(e) = B$ afterwards. Then $c$ is still a winner of the weighted election $(C, V_1 \cup V_0)$, and we obtain $\sum_{v_i \in V_0} w_i = B$ and thus $I \in$ 3-approval-PWUW-BW-RW-$\mathbb{N}$. ❑

From Theorems 3.1 and 5.6 we immediately obtain the following corollary.

**Corollary 5.7.** 3-approval-PWUW-BW-$\mathbb{N}$ *is in* P.

Alternatively, as pointed out by Zack Fitzsimmons and Edith Hemaspaandra in a personal communication, the result that PWUW-BW-RW-$\mathbb{N}$ is in P for 3-approval also follows immediately from Propositions 3.2 and 3.3 and the results by Fitzsimmons and Hemaspaandra [35]. They also note that since the dichotomy for the problem CCAV shown by Hemaspaandra, Hemaspaandra, and Schnoor [43] is exactly the same as for its succinct variant $\text{CCAV}_{\text{succ}}$ [35], it follows that the same dichotomy holds for all problems X such that CCAV $\leq_{\text{m}}^{\text{p}}$ X $\leq_{\text{m}}^{\text{p}}$ $\text{CCAV}_{\text{succ}}$. Hence, this immediately gives the

same dichotomy for PWUW-BW-RW-$\mathbb{N}$ (both its general and its succinct variant), for example.

Having pinpointed the complexity of our four problem variants for $k$-approval with $k \leq 3$, the following result now answers all remaining cases for $k$-approval with $k \geq 4$. Unlike for all previously considered problems, we now provide hardness results.

**Theorem 5.8.** *For each $k \geq 4$, $k$-approval-PWUW-BW-RW-$\mathbb{N}$ and $k$-approval-PWUW-BW-$\mathbb{N}$ are* NP*-complete.*

**Proof.** It is easy to see that both problems belong to NP: We can simply guess a weight allocation for the votes in $V_0$ satisfying all requirements (i.e., only a bound $B$ or a bound $B$ as well as some regions $R$) and then, for each solution guessed, we check efficiently whether the distinguished candidate is a winner of the weighted election $(C, V_1 \cup V_0)$.

To prove NP-hardness, we first give a proof for 4-approval-PWUW-BW-$\mathbb{N}$ by a polynomial-time many-one reduction from EXACT-COVER-BY-3-SETS and we then explain how to extend this proof to all values $k \geq 4$.

Let $(\mathscr{B}, \mathscr{S})$ be an X3C instance with $\mathscr{B} = \{b_1, \ldots, b_{3q}\}$ and $\mathscr{S} = \{S_1, \ldots, S_n\}$. We construct a $k$-approval-PWUW-BW-$\mathbb{N}$ instance $I = (C, V_1, V_0, c, B)$ with

- candidates $C = \{c, b_1, \ldots, b_{3q}, b_1^1, \ldots, b_{3q}^1, b_1^2, \ldots, b_{3q}^2, b_1^3, \ldots, b_{3q}^3\}$;
- $V_1$ having $q-1$ votes of the form $b_j > b_j^1 > b_j^2 > b_j^3 > \cdots$ for each $j$, $1 \leq j \leq 3q$;
- $V_0$ consisting of $n$ votes of the form $c > \overrightarrow{S_i} > \cdots$, $1 \leq i \leq n$;
- distinguished $c$ candidate; and
- $B = q$ being the bound on the total weight of the votes in $V_0$.

Recall that the votes in $V_1$ all have fixed weight one, and the weights of the votes in $V_0$ are from $\mathbb{N}$.

We show that $\mathscr{S}$ has an exact cover for $\mathscr{B}$ if and only if there are weights for the votes in $V_0$ satisfying the bound $B$ such that $c$ wins the weighted election $(C, V_1 \cup V_0)$.

From left to right, assume that there is an exact cover $\mathscr{S}' \subseteq \mathscr{S}$ for $\mathscr{B}$. By setting the weights of the votes $c > \overrightarrow{S_i} > \cdots$ to one for those $q$ subsets $S_i$ contained in $\mathscr{S}'$, and to zero for all other votes in $V_0$, $c$ is a winner of the election, as $c$ and all $b_j$, $1 \leq j \leq 3q$, receive exactly $q$ points, whereas $b_j^1$, $b_j^2$, and $b_j^3$, $1 \leq j \leq 3q$, receive $q-1$ points each.

Conversely, from right to left, assume that $c$ can be made a winner of the election by choosing the weights of the votes in $V_0$ appropriately. Note that the bound on the total weight for the votes in $V_0$ is $B = q$. Every $b_i$ gets $q-1$ points from the votes in $V_1$, and $c$ gets points only from the votes in $V_0$. Since there are always some $b_j$ getting points if a vote from $V_0$ has weight one, there are at least three $b_j$ having $q$ points if a vote from $V_0$ has weight one. Hence $c$ must get $q$ points from the votes in $V_0$ by setting the weight of $q$ votes to one. Furthermore, every $b_j$ can occur only once in the votes having weight one in $V_0$, as otherwise $c$ would not win. It follows that the $S_i$ corresponding to the votes of weight one in $V_0$ must form an exact cover for $\mathscr{B}$.

Hence, 4-approval-PWUW-BW-$\mathbb{N}$ is NP-complete. Applying Theorem 3.1, 4-approval-PWUW-BW-RW-$\mathbb{N}$ is NP-complete, too.

To extend the proof from $k = 4$ to all $k > 4$, we can add dummy candidates to fill the additional positions in the votes also receiving points. ❑

To summarize, we have shown that for $k \leq 3$ all four possible winner with uncertain weights variants can be solved in polynomial time. Furthermore, for $k \geq 4$, the variants PWUW-$\mathbb{N}$ and PWUW-RW-$\mathbb{N}$ belong to P as well, whereas the variants PWUW-BW-$\mathbb{N}$ and PWUW-BW-RW-$\mathbb{N}$ are NP-complete.

*5.2. k-Veto*

In this section, we focus on the voting rule $k$-veto. Recall that $k$-veto is a scoring protocol with scoring vector

$$\vec{\alpha} = (1, \ldots, 1, \underbrace{0, \ldots, 0}_{k}),$$

i.e., the last $k$ entries contain a 0, whereas all other entries contain a 1. Obviously, $k$-veto is a binary scoring protocol for every $k \in \mathbb{N}$, so Corollary 5.9 follows immediately from Theorem 3.4.

**Corollary 5.9.** *For $k \in \mathbb{N}$, $k$-veto-PWUW-RW-$\mathbb{N}$ and $k$-veto-PWUW-$\mathbb{N}$ belong to* P.

We now turn to the other two problem variants, starting with providing results for $k = 1$: veto-PWUW-BW-RW-$\mathbb{N}$ and veto-PWUW-BW-$\mathbb{N}$.

**Theorem 5.10.** *The problems* veto-PWUW-BW-RW-$\mathbb{N}$ *and* veto-PWUW-BW-$\mathbb{N}$ *are in* P.

**Proof.** First, we show that veto-PWUW-BW-RW-$\mathbb{N}$ is in P. Let $I = (C, V_1, V_0, c, B, R)$ be a modified veto-PWUW-BW-RW-$\mathbb{N}$ instance with $l_i = 0$ for all $R_i = [l_i, r_i]$, according to Lemma 5.1. We write $C = \{c_1, \ldots, c_{m-1}\} \cup \{c\}$ for the candidates. For every candidate $c_i$, $1 \leq i \leq m-1$, we determine

$$\mathit{diff}_{(C,V_1)}(c_i, c) = \mathit{score}_{V_1}(c_i) - \mathit{score}_{V_1}(c).$$

That is, $\mathit{diff}_{(C,V_1)}(c_i, c)$ describes how many more points $c_i$ receives in $V_1$ than $c$ does. In order for $c$ to be a winner of the weighted election $(C, V_1 \cup V_0)$, $c$ must obtain at least as many points as every other candidate in $C$. If we weigh a vote $v_j$ from $V_0$ with weight $w_j \in \mathbb{N}$, the relative difference of points does not change among all candidates but only with respect to the candidate being in the last position of $v_j$. We denote by

$$\psi = \{c_i \in C \mid \mathit{diff}_{(C,V_1)}(c_i, c) > 0\}$$

the set of all candidates obtaining more points in $V_1$ than $c$. To make $c$ a winner of the election by setting the weights of the votes in $V_0$ appropriately, for every candidate $c_i \in \psi$, we must identify votes $v_j \in V_0$ with $c_i$ in the last position such that the sum of the upper bounds of the corresponding intervals $R_j = [0, r_j]$ is greater than or equal

to $\mathit{diff}_{(C,V_1)}(c_i,c)$. If such votes do not exist in $V_0$, it follows immediately that $I$ is a NO-instance. Furthermore, it must hold that

$$\sum_{c_i \in \psi} \mathit{diff}_{(C,V_1)}(c_i,c) \leq B$$

is true, since otherwise $I$ is a NO-instance as well, as we cannot give more than $B$ points to $c$ via the votes from $V_0$.

If both assumptions are satisfied, we solve the instance as follows. For every $c_i \in \psi$, we distribute a total weight of $\mathit{diff}_{(C,V_1)}(c_i,c)$ to votes from $V_0$ with $c_i$ in the last position. Doing so, all candidates in $C \setminus \{c_i\}$ obtain $\mathit{diff}_{(C,V_1)}(c_i,c)$ points from these votes while $c_i$ receives 0 points. Hence, $c$ obtains as many points as $c_i$ from the votes in $V_1 \cup V_0$. We repeat this step for every $c_i \in \psi$. Afterwards, for all $c_i, 1 \leq i \leq m-1$, it holds that

$$\begin{aligned} & \mathit{score}_{V_1}(c) + \mathit{score}_{V_0}(c) - (\mathit{score}_{V_1}(c_i) + \mathit{score}_{V_0}(c_i)) \\ = \; & -\mathit{diff}_{(C,V_1)}(c_i,c) + \mathit{score}_{V_0}(c) - \mathit{score}_{V_0}(c_i) \\ = \; & -\mathit{diff}_{(C,V_1)}(c_i,c) + \mathit{diff}_{(C,V_1)}(c_i,c) \\ = \; & 0, \end{aligned}$$

so $c$ is a winner of the weighted election $(C, V_1 \cup V_0)$ and, consequently, $I$ is a YES-instance.

To see that veto-PWUW-BW-$\mathbb{N}$ is in P, too, simply use the reduction from Theorem 3.1 to reduce veto-PWUW-BW-$\mathbb{N}$ to veto-PWUW-BW-RW-$\mathbb{N}$. ❑

In order to prove that PWUW-BW-$\mathbb{N}$ and PWUW-BW-RW-$\mathbb{N}$ belong to P for 2-veto as well, we introduce another variant of the previously presented polynomial-time solvable GWBEM problem. According to Gabow [37] and Grötschel et al. [39, p. 259], the following variant of this problem is in P, too:[6]

| GENERALIZED-B-EDGE-COVER (GBEC) | |
|---|---|
| **Given:** | An undirected multigraph $G = (N, E)$ without loops, capacity-bounding functions $a_\ell, a_u \colon E \to \mathbb{N}$ and $b_\ell, b_u \colon N \to \mathbb{N}$, and a target integer $r \in \mathbb{N}$. |
| **Question:** | Is there a function $x \colon E \to \mathbb{N}$ with $\sum_{e \in E} x(e) \leq r$ such that for every edge $e \in E$ it holds that $a_\ell(e) \leq x(e) \leq a_u(e)$ and for every node $z \in N$ it holds that $b_\ell(z) \leq \sum_{e \in \delta(z)} x(e) \leq b_u(z)$, where $\delta(z)$ is the set of edges incident to node $z$? |

The difference to the earlier introduced GWBEM problem is that this time the weights assigned to the edges of $G$ are *not weighted* and we want the sum to be *at most* $r$ instead of *at least* $r$. Especially the last difference seems to be a bit subtle but is crucial for the upcoming proof of Theorem 5.11, as this time we work with a veto rule and, thus, construct the corresponding graph in such a way that positively weighted edges in the graph cause candidates to *not* obtain points.

[6] Again, Gabow [37] and Grötschel et al. [39, p. 259] formalize this problem variant as a minimization problem. Since this problem is polynomial-time solvable, the decision problem variant that we will use is in P as well.

**Theorem 5.11.** 2-veto-PWUW-BW-RW-$\mathbb{N}$ *and* 2-veto-PWUW-BW-$\mathbb{N}$ *belong to* P.

**Proof.** In order to prove that 2-veto-PWUW-BW-RW-$\mathbb{N}$ is in P, we reduce this problem to GBEC. Let $I = (C, V_1, V_0, c, B, R)$ be a modified 2-veto-PWUW-BW-RW-$\mathbb{N}$ instance with $R_i = [0, r_i]$ for $1 \leq i \leq |V_0|$, according to Lemma 5.1. We construct a GBEC instance $I' = (G, a_\ell, a_u, b_\ell, b_u, r)$ similar to the GWBEM instance in the proof of Theorem 5.6. We define the multigraph $G = (C \setminus \{c\}, E)$, where in order to specify $E$ we first define the set

$$V_0' = \{\cdots > x_1 > x_2 \in V_0 \mid \{x_1, x_2\} \cap \{c\} = \emptyset\}$$

consisting of all votes from $V_0$ with $c$ not being ranked in one of the last two positions. Then we define the edge set of $G$ as

$$E = \{\{x_1, x_2\} \mid \cdots > x_1 > x_2 \in V_0'\}.$$

Doing so, every edge in the graph corresponds to some vote in $V_0$, i.e., when we write $e_i$, we implicitly refer to the corresponding vote $v_i$ from $V_0$ (re-indexing the indices as needed). For every edge $e_i \in E$, we define $a_\ell(e_i) = 0$ and $a_u(e_i) = r_i$ for $R_i = [0, r_i]$. For every node $d \in C \setminus \{c\}$, we define

$$b_\ell(d) = \max\{0, score_{V_1}(d) - score_{V_1}(c)\}$$

and $b_u(d) = B$. Lastly, we define $r = B$. This completes the construction of $I'$, which can be realized in time polynomial in $|I|$.

To show that $I \in$ 2-veto-PWUW-BW-RW-$\mathbb{N}$ holds if and only if $I' \in$ GBEC holds follows a similar approach as the one for Theorem 5.6.

From left to right, assume $I \in$ 2-veto-PWUW-BW-RW-$\mathbb{N}$. Then there exist weights $w_i \in \mathbb{N}$ for $1 \leq i \leq |V_0|$ such that $c$ wins the weighted election $(C, V_1 \cup V_0)$ while $w_i \in R_i = [0, r_i]$ and $\sum_{i=1}^{|V_0|} w_i \leq B$ hold. Now, for every edge $e_i \in E$, we set $x(e_i) = w_i$. Consequently, $a_\ell(e_i) = 0 \leq x(e_i) = w_i \leq a_u(e_i) = r_i$ is satisfied for all edges $e_i \in E$. Furthermore, we obtain

$$\sum_{e_i \in E} x(e_i) = \sum_{e_i \in E} w_i = \sum_{v_i \in V_0'} w_i \leq B = r.$$

Since $c$ is a winner of the weighted election $(C, V_1 \cup V_0)$, for all $d \in C \setminus \{c\}$ it holds that

$$score_{V_1}(c) + score_{V_0}(c) \geq score_{V_1}(d) + score_{V_0}(d).$$

Hence, $c$ must have gained at least $\max\{0, score_{V_1}(d) - score_{V_1}(c)\}$ more points than $d$ from the weighted votes in $V_0$. The only possibility to not obtain a point from a positively weighted vote in $V_0$ is to be in one of the last two positions. Consequently, $d$ must have been in one of the two last positions for at least $\max\{0, score_{V_1}(d) - score_{V_1}(c)\}$ positively weighted votes in $V_0$ and thus, it holds that

$$b_\ell(d) = \max\{0, score_{V_1}(d) - score_{V_1}(c)\} \leq \sum_{e_i \in \delta(d)} x(e_i) = \sum_{e_i \in \delta(d)} w_i \leq B = b_u(d).$$

Hence, all requirements are satisfied and $I' \in$ GBEC is true.

From right to left, suppose $I' \in$ GBEC. Then there exists a function $x\colon E \to \mathbb{N}$ such that

(i) $\sum_{e\in E} x(e) \leq r$ is satisfied,
(ii) for all $e \in E$, it holds that $a_\ell(e) \leq x(e) \leq a_u(e)$, and
(iii) for all $d \in C \setminus \{c\}$, it holds that $b_\ell(d) \leq \sum_{e\in\delta(d)} x(e) \leq b_u(d)$.

For every edge $e_i \in E$, we set the weight of the corresponding vote $v_i \in V_0$ to $w_i = x(e_i)$. For all remaining votes in $V_0$, we set the weight to 0. Consequently, we satisfy $w_i \in R_i = [0, r_i]$ for every vote $v_i \in V_0$, since

$$a_\ell(e_i) = 0 \leq x(e_i) = w_i \leq a_u(e_i) = r_i$$

holds according to (ii). Furthermore, according to (iii), every candidate $d \in C \setminus \{c\}$ satisfies

$$\begin{aligned} b_\ell(d) &= \max\{0, score_{V_1}(d) - score_{V_1}(c)\} \\ &\leq \sum_{e\in\delta(d)} x(e) \leq b_u(d) = B. \end{aligned} \tag{32}$$

The overall sum of points allocated to $c$ by the positively weighted votes from $V_0$ corresponds to $score_{V_0}(c) = \sum_{e\in E} x(e)$, as $c$ never occurs in one of the last two positions for the positively weighted votes in $V_0$. The overall sum of points allocated to every other candidate $d \in C \setminus \{c\}$ by the votes in $V_0$ corresponds to

$$score_{V_0}(d) = \sum_{e\in E} x(e) - \sum_{e\in\delta(d)} x(e).$$

Hence, we obtain

$$\sum_{e\in\delta(d)} x(e) = \sum_{e\in E} x(e) - score_{V_0}(d) = score_{V_0}(c) - score_{V_0}(d).$$

Together with (32) this yields

$$\max\{0, score_{V_1}(d) - score_{V_1}(c)\} \leq score_{V_0}(c) - score_{V_0}(d),$$

which implies

$$score_{V_1}(d) - score_{V_1}(c) \leq score_{V_0}(c) - score_{V_0}(d),$$

which in turn is equivalent to

$$score_{V_1}(d) + score_{V_0}(d) \leq score_{V_1}(c) + score_{V_0}(c),$$

so $c$ is a winner of the weighted election $(C, V_1 \cup V_0)$.

Finally, from (i) we know that

$$\sum_{e_i\in E} x(e_i) = \sum_{e_i\in E} w_i = \sum_{v_i\in V_0} w_i \leq r = B.$$

Consequently, $I \in$ 2-veto-PWUW-BW-RW-$\mathbb{N}$ is true and it follows that 2-veto-PWUW-BW-RW-$\mathbb{N}$ is in P.

With the reduction from Theorem 3.1 it immediately follows that PWUW-BW-$\mathbb{N}$ is in P, too. ❑

The only cases left are $k$-veto-PWUW-BW-RW-$\mathbb{N}$ and $k$-veto-PWUW-BW-$\mathbb{N}$ for $k \geq 3$. For these problems, we establish hardness results.

**Theorem 5.12.** *For each $k \geq 3$, $k$-veto-PWUW-BW-$\mathbb{N}$ and $k$-veto-PWUW-BW-RW-$\mathbb{N}$ are* NP*-complete.*

**Proof.** Membership in NP is obvious for both problems: We can simply guess a weight allocation for the votes in $V_0$ and then, for each solution guessed, we check whether this allocation satisfies all requirements and makes $c$ a winner of the weighted election.

In order to prove NP-hardness, we first give a reduction from X3C to 3-veto-PWUW-BW-$\mathbb{N}$. Afterwards, we will explain how to obtain NP-hardness for 3-veto-PWUW-BW-RW-$\mathbb{N}$ and how to extend these proofs to all values $k \geq 3$.

Let $I = (\mathscr{B}, \mathscr{S})$ with $\mathscr{B} = \{b_1, \ldots, b_{3q}\}$ and $\mathscr{S} = \{S_1, \ldots, S_n\}$ be a given X3C instance. We construct a 3-veto-PWUW-BW-$\mathbb{N}$ instance $I' = (C, V_1, V_0, c, B)$ as follows. Define $C = \{c, d_1, d_2, d_3\} \cup \mathscr{B}$, where $d_1$, $d_2$, and $d_3$ are dummy candidates required to construct the votes in $V_1$, and set $B = q$. Define $V_1$ to consist of one vote of the form

$$u_1 = \overrightarrow{\mathscr{B}} > d_1 > d_2 > d_3 > c$$

and two votes of the form

$$u_j = \overrightarrow{\mathscr{B}} > c > d_1 > d_2 > d_3, \quad j \in \{2,3\}.$$

Doing so, the candidates obtain the following scores from the votes in $V_1$:

| | $b_i \in \mathscr{B}$ | $c$ | $d_1$ | $d_2$ | $d_3$ |
|---|---|---|---|---|---|
| $score_{V_1}(\cdot)$ | 3 | 2 | 1 | 0 | 0 |

Next, define $V_0$ to consist of votes of the form

$$v_i = \overrightarrow{\mathscr{B} \setminus S_i} > d_1 > d_2 > d_3 > c > \overrightarrow{S_i}$$

for $1 \leq i \leq m$. Obviously, this construction of $I'$ is possible in time polynomial in $|I|$.

Note that no matter how many votes from $V_0$ we weigh positively, $c$ and all three dummy candidates—$d_1$, $d_2$, and $d_3$—always obtain the same number of points. Thus $c$ always beats all three dummy candidates in the weighted election $(C, V_1 \cup V_0)$.

We now show that $I \in$ X3C holds if and only if $I' \in$ 3-veto-PWUW-BW-RW-$\mathbb{N}$ holds.

From left to right, assume that $I \in$ X3C. That is, there exists an exact cover $\mathscr{S}' \subseteq \mathscr{S}$ with $|\mathscr{S}'| = q$ such that $\bigcup_{S_i \in \mathscr{S}'} S_i = \mathscr{B}$. For every $S_i \in \mathscr{S}'$, we set the weight of the corresponding vote $v_i \in V_0$ to $w_i = 1$ and for all remaining votes in $V_0$ to 0. Hence, $\sum_{i=1}^{|V_0|} w_i = q = B$ is satisfied.

Since $\mathscr{S}'$ is an exact cover, it holds that every $b_i \in \mathscr{B}$ appears exactly once behind $c$ in the positively weighted votes from $V_0$, so $c$ gains one point in the corresponding vote, whereas $b_i$ obtains 0 points. Consequently, the candidates' scores for the weighted election are as follows:

| | $b_i \in \mathscr{B}$ | $c$ | $d_1$ | $d_2$ | $d_3$ |
|---|---|---|---|---|---|
| $score_{(C, V_1 \cup V_0)}(\cdot)$ | $q+2$ | $q+2$ | $q+1$ | $q$ | $q$ |

Thus $c$ is a winner of the weighted election $(C, V_1 \cup V_0)$, so $I' \in$ 3-veto-PWUW-BW-$\mathbb{N}$.

From right to left, assume that $I' \in$ 3-veto-PWUW-BW-$\mathbb{N}$. Then there exist weights $w_i$ for $1 \leq i \leq |V_0|$ such that $c$ is a winner of the weighted election $(C, V_1 \cup V_0)$. Since $B = q$, the sum of the weights for the votes in $V_0$ cannot be larger than $q$. Accordingly, we study the following two cases.

**Case 1:** $\sum_{i=1}^{|V_0|} w_i < q$. If the total sum of all weights over $V_0$ is less than $q$, not all candidates in $\mathscr{B}$ can appear behind $c$ in the positively weighted votes from $V_0$. However, appearing behind $c$ in the positively weighted votes in $V_0$ is the only chance for $c$ to win against a candidate from $\mathscr{B}$. Consequently, there must exist at least one candidate $b_i \in \mathscr{B}$ who beats $c$. This contradicts the assumption that $c$ is a winner of the weighted election $(C, V_1 \cup V_0)$, making this case impossible.

**Case 2:** $\sum_{i=1}^{|V_0|} w_i = q$. If there is a weight $w_i > 1$ for a vote from $V_0$, the same argument as in Case 1 holds for the remaining $3(q-1)$ candidates and remaining weight of at most $q-2$. Hence, for all weights it must hold that $w_i \leq 1$. Then every candidate from $\mathscr{B}$ can occur behind $c$ in the positively weighted votes from $V_0$ at least once. There cannot be a candidate occurring more than once behind $c$, since then another candidate could not occur at all behind $c$. This would contradict our assumption of $c$ winning the weighted election in the same way as in the first case. Therefore, every candidate from $\mathscr{B}$ must occur *exactly* once behind $c$ in the positively weighted votes from $V_0$. Selecting these positively weighted votes from $V_0$ yields an exact cover, so $I$ is a YES-instance.

This concludes the reduction and the NP-hardness proof for 3-veto-PWUW-BW-$\mathbb{N}$. By using $k > 3$ instead of three dummy candidates in this reduction, we obtain that $k$-veto-PWUW-BW-$\mathbb{N}$ is NP-hard for $k > 3$ as well. Finally, applying Theorem 3.1 it follows that for $k \geq 3$ $k$-veto-PWUW-BW-RW-$\mathbb{N}$ is NP-hard, and hence, NP-complete, too. ❑

Note, that by the reduction CCAV $\leq_{\mathrm{m}}^{\mathrm{p}}$ PWUW-BW-RW-$\mathbb{N}$ from Proposition 3.2 combined with the results by Lin [49], who has shown that $k$-veto-CCAV is NP-hard for $k \geq 3$, we obtain an alternative proof to the previous one to show that $k$-veto-PWUW-BW-RW-$\mathbb{N}$ is NP-complete. With this observation we conclude our study of the scoring protocol $k$-veto, as we have settled the complexity for all four problem variants of the possible winner with uncertain weights problem for all $k \geq 1$.

### *5.3. Plurality with Runoff*

For plurality with runoff, we have seen in Section 4 that all four variants of the possible winner with uncertain weights problem are in P when the weights have nonnegative rational values. For nonnegative integer weights, we now solve all open questions regarding the computational complexity of these problems.

In Proposition 3.3, we have shown that PWUW-BW-RW-$\mathbb{N} \leq_{\mathrm{m}}^{\mathrm{p}}$ $\mathrm{CCAV}_{\mathrm{succ}}$ for every voting rule. Recall that the succinct representation in a $\mathrm{CCAV}_{\mathrm{succ}}$ instance means that identical votes are not listed one by one but just once along with a binary number giving the multiplicity of this vote. Together with the facts from Theorem 3.1 that

PWUW-BW-$\mathbb{N} \leq_m^p$ PWUW-BW-RW-$\mathbb{N}$ and PWUW-RW-$\mathbb{N} \leq_m^p$ PWUW-BW-RW-$\mathbb{N}$, it follows that if $\text{CCAV}_{\text{succ}}$ for plurality with runoff is in P, each of PWUW-BW-RW-$\mathbb{N}$, PWUW-BW-$\mathbb{N}$, and PWUW-RW-$\mathbb{N}$ is in P for plurality with runoff as well. Erdélyi et al. [30] showed that CCAV (in standard representation) is in P for plurality with runoff. Alas, their approach cannot be easily adapted for succinct representation as their algorithm iterates over all possible values $\ell' \leq \ell$ and $\ell$ is not polynomially bounded in succinct representation. Instead, after some preprocessing steps we will solve the problem with an integer linear program (ILP) in the following theorem, similarly to how Fitzsimmons and Hemaspaandra [35] have handled election problems in succinct representation. Regarding tie-breaking, we assume that whenever a tie occurs, we can freely choose how it is broken.

**Theorem 5.13.** Plurality-with-runoff-$\text{CCAV}_{\text{succ}}$ *is in* P.

**Proof.** Let $I = (C, c, V, W, \ell)$ be a plurality-with-runoff-$\text{CCAV}_{\text{succ}}$ instance. To determine whether $I$ is a YES- or a NO-instance, we proceed as follows. We execute the algorithm to be described below for every candidate $d \in C \setminus \{c\}$, checking whether we can choose at most $\ell$ votes from $W$ such that $d$ and $c$ enter the runoff and $c$ wins it. Note that the algorithm for any single candidate $d$ runs in polynomial time, so we can execute it for all candidates in $C \setminus \{c\}$ in sequence while staying in time polynomial in $|I|$. If our algorithm is successful for some candidate $d \neq c$, we know that $I$ is a YES-instance; if it fails for every candidate $d \in C \setminus \{c\}$, we know that $I$ is a NO-instance.

Let us begin to describe the algorithm for a single candidate $d \in C \setminus \{c\}$. The following steps try to gradually build a set of votes $V'$ with $V \subseteq V'$ and $0 \leq |W \cap V'| \leq \ell$, so that $c$ and $d$ reach the runoff which, in turn, is won by $c$. After every step we have three options: (1) we have reached our goal, i.e., $c$ wins the runoff, (2) our goal is not reachable with the current candidate $d$, i.e., we terminate the algorithm at this point and proceed with the next candidate, or (3) it is not yet determined whether $c$ can win and we continue with the next step.

In a first step, we select $\ell_c^1 = \max_{a \in C \setminus \{c,d\}} \mathit{score}_V(a) - \mathit{score}_V(c)$ votes with $c$ on top from $W$ and add them to our new set of overall votes $V'$. Analogously, we select $\ell_d^1 = \max_{a \in C \setminus \{c,d\}} \mathit{score}_V(a) - \mathit{score}_V(d)$ votes with $d$ on top from $W$ and add these to $V'$. If $\ell_c^1 < 0$ (respectively, $\ell_d^1 < 0$) we know that $c$ (respectively, $d$) already reaches the runoff so we add no votes with $c$ (respectively, with $d$) on top from $W$. Doing so, we ensure that $c$ and $d$ have a score as least as high as all other candidates in $C$ in the first round. Of course, if there are not sufficiently many votes in $W$ available, we skip $d$ at this point. Assuming some tie-breaking in favor of $c$ and $d$, it follows that $c$ and $d$ enter the runoff. Then we calculate $\mathit{diff}_{(\{c,d\},V')}(d,c)$, i.e., the score difference of $d$ and $c$ in the runoff. If $\mathit{diff}_{(\{c,d\},V')}(d,c) \leq 0$, it follows that $c$ wins the runoff and we're done, i.e., $I$ is a YES-instance. Otherwise, if $\mathit{diff}_{(\{c,d\},V')}(d,c) > 0$, it follows that $c$ does not yet win the runoff. Consequently, with the next steps we add additional votes to $V'$ in such a way that $c$ obtains enough points in the runoff to beat $d$ if that is possible.

In a second step, we therefore add all votes with $c$ on top in $W \setminus V'$ to $V'$. Surely, this is the best way to influence the score difference of $c$ and $d$ in the runoff in favor of $c$. Doing so, note that afterwards there are no further votes with $c$ on top left in $W$, so $\xi_{\max}^c = \mathit{score}_{V'}(c)$ is the final score of $c$ in the first round from this point on. Again, we check if $\mathit{diff}_{(\{c,d\},V')}(d,c) \leq 0$ and proceed to the next step if this is not the case.

In a third step, let $\kappa = \min\{\xi^c_{\max}, score_{V'}(d)\}$ and denote by $w_a^{c>d}$, for every $a \in C \setminus \{c,d\}$, the number of votes in $W$ with $a$ on top and $c$ in front of $d$. For every $a \in C \setminus \{c,d\}$, we can add $\min\{w_a^{c>d}, \kappa - score_{V'}(a)\}$ votes with $a$ on top and $c$ before $d$ from $W$ to $V'$. Doing so, we add points for $c$ in the runoff, while ensuring that $c$ and $d$ still enter the runoff. That is the case because we add only as many votes from $W$ with $a$ on top to $V'$ that afterwards $a$'s score is at most $\kappa$, which is less than or equal to the scores of $c$ and $d$ in the first round. Again, we check whether $c$ wins the runoff against $d$. If that is the case, we have a YES-instance. Otherwise, we continue with step four.

Step four is a preparing step ahead of the final—the fifth—step. Note that if $\kappa = \xi^c_{\max}$ would hold, adding further votes from $W \setminus V'$ to $V'$ that have some $a \in C \setminus \{c,d\}$ on top and $c$ before $d$ would mean that either $c$ or $d$ does not join the runoff and we can restart the process with the next candidate. Thus we can assume that $\kappa = score_{V'}(d) < \xi^c_{max}$. Now, we know that for all candidates $a \in C \setminus \{c,d\}$ it holds that $score_{V'}(a) = \kappa$ or there are no more votes left in $W \setminus V'$ with $a$ on top and $c$ in front of $d$. For every $a \in C \setminus \{c,d\}$, let $\widehat{w}_a^{c>d}$ be the number of votes left in $W \setminus V'$ with $a$ on top and $c$ in front of $d$ and let $\widehat{w}_d$ be the number of votes left in $W \setminus V'$ with $d$ on top. Denote by

$$C' = \{a \in C \setminus \{c,d\} \mid \widehat{w}_a^{c>d} > 0\}$$

the set of candidates other than $c$ or $d$ for which we can still add further votes to $V'$ that have a candidate from $C'$ on top and rank $c$ in front of $d$. Note that all candidates $a \in C \setminus \{c,d\}$ with $a \notin C'$ are irrelevant from this point on, as they are beaten or tied by $c$ and $d$ and we have no reason to add votes that increase their score because those votes (if there exist any) rank $d$ in front of $c$. Of course, the previous steps can be done only as long as the total sum of all added votes is smaller than $\ell$. If we would reach $\ell$ at some point, we immediately terminate the addition of votes and check whether $c$ wins the election.

Thus, entering the final step, we denote by $\ell' > 0$ the number of votes left to be added. Furthermore, sort $C' = \{a_1, \ldots, a_m\}$ ascendingly regarding $\widehat{w}_{a_i}^{c>d}$, $1 \leq i \leq m$. We now design an ILP that can be used to determine whether $c$ can win against $d$ in the runoff while adding no more than $\ell'$ further votes from $W \setminus V'$ to $V'$. The ILP uses the two variables, $\ell_d$ and $\ell_{other}$, representing the number of votes added with $d$ or, respectively, with $a \notin \{c,d\}$ on top and $c$ before $d$ from $W \setminus V'$ to $V'$, and consists of the following five constraints:

- Obviously, we cannot add more votes from $W \setminus V'$ to $V'$ with $d$ on top than there are left, i.e., we demand

$$\ell_d \leq \widehat{w}_d. \tag{33}$$

- Next, we want to ensure that $c$ and $d$ enter the runoff. As discussed earlier, we already added all votes with $c$ on top to $V'$. Hence, the following constraint ensures that $d$ has a score at most as high as the score of $c$, which, combined with the last constraint, enforces all candidates $a \notin \{c,d\}$ to have a score at most

as high as $c$ as well:[7]

$$\ell_d \leq \xi^c_{\max} - score_{V'}(d). \tag{34}$$

- The next constraint ensures that the number of votes with a candidate different from $c$ and $d$ on top and $c$ before $d$ added to $V'$ is high enough to ensure that $c$ beats $d$ in the runoff. This works because every vote with $a \notin \{c,d\}$ on top and $c$ before $d$ added to $V'$ increases $c$'s runoff score by one and keeps $d$'s runoff score constant,

$$\mathit{diff}_{(\{c,d\},V')}(d,c) + \ell_d \leq \ell_{\text{other}}. \tag{35}$$

- Of course, we can only add as many votes to $V'$ in total as we are allowed by the original instance. Hence, the next constraint ensures that we stay within this bound,

$$\ell_{\text{other}} + \ell_d \leq \ell'. \tag{36}$$

- Finally, the number of votes with $a \notin \{c,d\}$ on top and $c$ before $d$ is not arbitrarily large. Thus, for every candidate $a \in C'$, we can add at most $\widehat{w}_a^{c>d}$ votes to $V'$. At the same time, we must ensure that $d$ enters the runoff. Hence, the number of votes with $a$ on top added is also limited by the number of votes added with $d$ on top,

$$\ell_{\text{other}} \leq \sum_{i=1}^{m} \min\{\widehat{w}_{a_i}^{c>d}, \ell_d\}. \tag{37}$$

  This constraint enforces $d$ to enter the runoff together with $c$. The idea is that if $\ell_{other}$ is smaller than or equal to the sum of these minima, we can assign to every candidate in $C'$ some number of appropriate votes from $W \setminus V'$ according to her corresponding minimum. With this allocation we ensure that we do not allocate more votes than there exist for a given candidate $a \notin \{c,d\}$ and at the same time we ensure that the score of $a$ in the first round is at most as high as $d$'s score. Since $d$'s score is at most as high as $c$'s score, it follows that $c$ and $d$ enter the runoff.

It is easy to see that the above constraints are satisfied if and only if it is possible to add at most $\ell'$ unregistered votes from $W \setminus V'$ to $V'$ such that $c$ wins the runoff against $d$.

This ILP has a constant number of variables, so it can be solved by the means presented in the work of Lenstra [48]. However, to do so, we must explain how one handles the last constraint regarding the minima. We loosely follow the ideas due to Fitzsimmons and Hemaspaandra [35]. As noted earlier, the candidates in $C'$ are sorted ascendingly with respect to $\widehat{w}_{a_i}^{c>d}$, $1 \leq i \leq m$. Thus, instead of solving one ILP, we

[7] It is important for this constraint that if we run the ILP, we know that $\xi^c_{max} > score_{V'}(d)$.

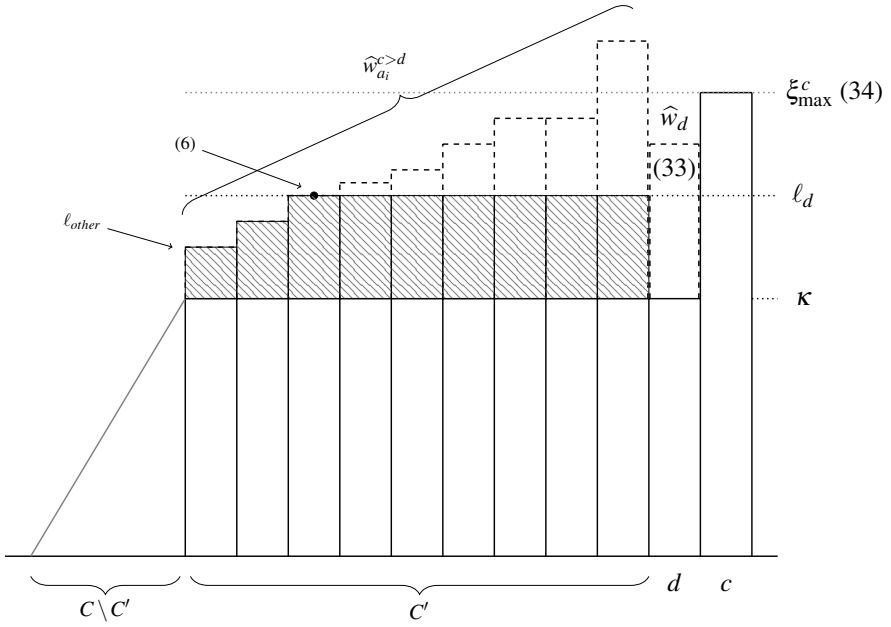



Figure 2: Sketch depicting the ILP constructed after the fourth step. The value of $\ell_d$ is here illustrated as a potential value and the numbers in brackets refer to the corresponding constraints. The hatched area corresponds to $\ell_{other}$ for the given $\ell_d$ .

in fact solve $|C'|$ ILPs. For the first ILP, we assume for all candidates $a_i \in C'$ that $\ell_d = \min\{\widehat{w}_{a_i}^{c>d}, \ell_d\}$. For each $i$ with $1 \leq i \leq |C'|$, in the $i$-th ILP we assume that for all $i' < i$, it holds that $\widehat{w}_{a_{i'}}^{c>d} = \min\{\widehat{w}_{a_{i'}}^{c>d}, \ell_d\}$, and for all $i''$ with $i \leq i'' \leq |C'|$, it holds that $\ell_d = \min\{\widehat{w}_{a_{i''}}^{c>d}, \ell_d\}$. Doing so, we replace the sum over the minima appropriately for each ILP. In addition, for the $i$-th ILP, $1 \leq i \leq |C'|$, we add a sixth constraint:

$$\widehat{w}_{a_{i-1}}^{c>d} \leq \ell_d \leq \widehat{w}_{a_i}^{c>d}, \tag{38}$$

assuming $\widehat{w}_{a_0}^{c>d} = 0$. This prevents the corresponding ILP from choosing a value for $\ell_d$ that does not correspond to the chosen min-values of constraint (37). This approach is illustrated in Figure 2. Afterwards, we solve all these constructed ILPs and check for the feasible ones. If we obtain at least one feasible ILP, it follows that $c$ can win against $d$ in the runoff, so $I$ is a YES-instance. Otherwise, if all ILPs are infeasible, it follows that $c$ cannot win against $d$ and we go to the next candidate; if $c$ fails against all candidates $d \in C \setminus \{c\}$, $I$ is a NO-instance.

Finally, we discuss how we handle lists of votes in succinct representation in the algorithm above. Of course, $V'$ is then also given in succinct representation, and whenever we have said to "add votes" from one list in succinct representation to $V'$, we mean that for every vote type of the votes to add we increase its binary number in $V'$ by the value to add, thus adding votes in bulk instead of one by one. Similarly, when performing difference operations $A \setminus B$ of lists of votes in succinct representation, we

decrease a vote types' number in $A$ by the vote types' number in $B$, keeping it at least zero. ❑

Let us briefly discuss the underlying idea that helped us to design and execute this proof. Once the fourth step of the proof is done, and we prepare everything for the ILP to be solved, we named the idea for the upcoming part as the *buoy principle*. Basically, we imagined the ILP to take place in an empty box of height $\xi^c_{max}$ that is slowly filled with water. Furthermore, we thought of $\ell_d$ as a buoy leashed by a chain of length $\hat{w}_d$ to the ground of the box. The water level then described the current value of $\ell_{other}$, taking the buoy $\ell_d$ with upwards. The amount of water is not infinite but corresponds to the maximum value $\ell_{other}$ could take. Slowly filling the box with water, we increase $\ell_{other}$ in order to improve $c$'s runoff score. However, as $\ell_d$ rises with the water level, we ensure that $d$ enters the runoff together with $c$ and no other candidate kicks $d$ out of the runoff. At some point either there is no more water left to be added to the box, i.e., we added as many votes as possible to $V'$ with $c$ before $d$, the box is completely filled, i.e., adding more votes to $V'$ would prevent $c$ from joining the runoff, or the buoy cannot raise anymore and would drown, i.e., adding more votes to $V'$ would prevent $d$ from joining the runoff. At this point, we know that we cannot find a better constellation of votes to be added to $V'$ in favor of $c$ to potentially win the runoff and all that is left to do is to check whether $c$ in this resulting setting indeed wins the runoff. We think that this idea of the buoy principle might be useful in other settings that try to solve similar problems as well. Using the metaphor of a buoy in a box slowly filling with water helps to imagine the most important—even though, of course, not every—constraint of such an ILP and helps to write down all formal conditions of the ILP.

Combining the initial argument at the beginning of this section with Theorem 5.13, we immediately obtain the following corollary.

**Corollary 5.14.** *For plurality with runoff,* PWUW-BW-RW-$\mathbb{N}$*,* PWUW-RW-$\mathbb{N}$*, and* PWUW-BW-$\mathbb{N}$ *are in* P.

Now, having this corollary and the previous proof, the only question left open for plurality with runoff regards the computational complexity of PWUW-$\mathbb{N}$. We answer this question with the next theorem.

**Theorem 5.15.** *For plurality with runoff,* PWUW-$\mathbb{N}$ *is in* P.

**Proof.** For plurality with runoff, a given instance $I = (C, V_1, V_0, c)$ of PWUW-$\mathbb{N}$ can be solved in polynomial time as follows. If there is a vote in $V_0$ with $c$ on top, we have a YES-instance, as we can simply assign this vote a large enough weight (e.g., $|V_1|$) and all other votes a weight of zero, so that $c$ wins the runoff. Otherwise, we define a PWUW-RW-$\mathbb{N}$ instance $I' = (C, V_1, V_0, c, R)$ (for plurality with runoff), with each region $R_i \in R$ ranging from zero to $|V_0||V_1|$. We can solve this instance in polynomial time, see Corollary 5.14, and then exploit the fact that $I$ is a YES-instance of PWUW-$\mathbb{N}$ if and only if $I'$ is a YES-instance of PWUW-RW-$\mathbb{N}$. To see this, we show that we can never assign a weight greater than $|V_0||V_1|$ to a vote in $V_0$ if $c$ wins. Assume for a contradiction that a vote $v \in V_0$ was given a weight greater than $|V_0||V_1|$ and $c$ won the runoff. Let $d$ be the candidate on top of $v$. We know that none of the votes in $V_0$ has

$c$ on top; otherwise, we would already be done. So $c$ has a score of at most $|V_1|$ in the first round. As $c$ reaches the runoff and the score of $d$ is greater than $c$'s score, all other candidates $C \setminus \{c,d\}$ have at most the same score as $c$. Thus the weight of each other vote in $V_0 \setminus \{v\}$ that does not have $d$ on top is at most $|V_1|$ (i.e., the upper bound on $c$'s score). Then the score $c$ gains in the runoff from the eliminated candidates is at most $(|V_0|-1)|V_1|$ which sums up to an upper bound of $|V_0||V_1|$ of $c$'s score in the runoff. But $d$'s score is greater than $|V_0||V_1|$, since the vote $v$ with weight greater than $|V_0||V_1|$ has $d$ on top, so $c$ loses the runoff, which is a contradiction. Note that there might be other votes in $V_0$ that have $d$ on top, but their weight is irrelevant for our argument, as $d$ already wins the first round and the runoff even without additional points from them. ❑

*5.4. Veto with Runoff*

Turning now to veto with runoff, we investigate our four PWUW variants for nonnegative integer weights. Again, as for plurality with runoff, we assume that if any ties occur, we can freely choose how to break these. Furthermore, we also make use of Proposition 3.3, which said that PWUW-BW-RW-$\mathbb{N}$, PWUW-BW-$\mathbb{N}$, and PWUW-RW-$\mathbb{N}$ reduce to $\text{CCAV}_{\text{succ}}$ (i.e., for instances in succinct representation) for any voting rule. For veto with runoff, CCAV in standard representation was shown to belong to P by Erdélyi et al. [30], but we run into the same issue as for plurality with runoff when we try to adapt their proof to obtain a reduction for instances in succinct representation. Therefore, we begin by proving that $\text{CCAV}_{\text{succ}}$ can be solved in polynomial time for veto with runoff, too.

**Theorem 5.16.** *For* veto with runoff, $\text{CCAV}_{\text{succ}}$ *is in* P.

**Proof.** Let $I = (C,c,V,W,\ell)$ be a CCAV instance in succinct representation. We proceed as follows to determine whether $I$ is a YES- or a NO-instance.

The procedure we describe afterwards runs in time polynomial in $|I|$, so we can execute it for all candidates $d \in C \setminus \{c\}$ in sequence. Furthermore, we emphasize that there is no situation where it would make sense to add votes from $W$ that veto $c$. Let $V'$ denote the final list of votes with $V \subseteq V'$ and $0 \leq |W \cap V'| \leq \ell$ constructed in the following steps.

We proceed as follows to ensure that $c$ and $d$ enter the runoff. Let $\kappa = \min\{score_V(c), score_V(d)\}$. For every candidate $a \in C \setminus \{c,d\}$ with $score_V(a) > \kappa$, we must add $score_V(a) - \kappa$ votes from $W$ to $V'$ that veto $a$. If there are not enough such votes in $W$ or we cannot add them due to the limit $\ell$, either $c$ or $d$ cannot reach the runoff. In this case, we can skip $d$ and proceed with the next candidate from $C \setminus \{c\}$. Otherwise (i.e., if there are enough such votes in $W$ that we can add), check whether $diff_{(\{c,d\},V')}(c,d) \geq 0$ is true for the current $V'$. In this case, $c$ beats or ties with $d$ in the runoff, and it follows that $I$ is a YES-instance. If this is not the case, we must add more votes to $V'$ that rank $c$ in front of $d$. We can only add at most

$$\min_{a \in C \setminus \{c,d\}} diff_{(C,V')}(d,a)$$

votes vetoing $d$ from $W \setminus V'$ to $V'$. Adding more votes would cause $d$ to not enter the runoff. If this is not enough to make up the deficit in the runoff, i.e., if

$$\mathit{diff}_{(\{c,d\},V')}(c,d) < 0$$

is still true for the current $V'$, we must add $|\mathit{diff}_{(\{c,d\},V')}(c,d)|$ votes from $W \setminus V'$ to $V'$ which veto some candidate other than $c$ and $d$ and rank $c$ before $d$. At this point, it does not matter which votes we add since those votes only hurt the vetoed candidate in the first round, which is neither $c$ nor $d$. If there are not enough votes of this type in $W \setminus V'$, we cannot make $c$ win against $d$ in the runoff. Hence, we must proceed with the next candidate. Otherwise, $c$ wins the runoff and, thus, it follows that $I$ is a YES-instance. If, at any point, we have reached the limit $\ell$ of added votes, we stop the process and check if we have successfully made $c$ a winner or skip to the next candidate. If we iterated over all candidates without finding a candidate against which $c$ wins the runoff, $I$ is a NO-instance. For a discussion on how lists of votes in succinct representation are handled see the end of the proof of Theorem 5.13. ❑

We immediately have the following corollary.

**Corollary 5.17.** *For veto with runoff, the problems* PWUW-RW-$\mathbb{N}$*,* PWUW-BW-$\mathbb{N}$*, and* PWUW-BW-RW-$\mathbb{N}$ *are in* P.

The only open case for veto with runoff is PWUW-$\mathbb{N}$, which we consider now.

**Theorem 5.18.** *For veto with runoff,* PWUW-$\mathbb{N}$ *is in* P.

**Proof.** Let $I = (C, V_1, V_0, c)$ be a PWUW-$\mathbb{N}$ instance for veto with runoff. We assume (just as Erdélyi et al. [30] do) that ties are always broken in favor of $c$. For $d \in C$, denote by $D_d^\star = \{r \in C \mid \mathit{score}_{V_1}(r) \star \mathit{score}_{V_1}(d)\} \subseteq C$ the set of candidates from $C$ satisfying the relation determined by $\star$ (e.g., $>$) with $d$ regarding the scores in $V_1$. We distinguish the following three cases:

**Case 1:** Assume that there are at least two candidates $d_1, d_2 \in D_c^>$ without votes vetoing them in $V_0$. In this case, we cannot allocate weights to the votes in $V_0$ such that $c$ has a veto score at least as high as one of the two candidates. Hence, $c$ will never enter the runoff, so $I$ is a NO-instance.

**Case 2:** Assume that there is only one candidate $d_1 \in D_c^>$ without votes vetoing her in $V_0$. In this case, we can allocate weights to the votes in $V_0$ such that $c$ enters the runoff together with $d_1$. In order to achieve this, for every candidate $d_2 \in D_c^> \setminus \{d_1\}$, allocate weight $\mathit{score}_{V_1}(d_2) - \mathit{score}_{V_1}(c)$ to a vote that vetoes $d_2$ in $V_0$. Afterwards, $c$ has a score at least as high as every other candidate in $D_c^> \setminus \{d_1\}$ and, thus, reaches the runoff. Now, if $c$ beats or ties $d_1$ in the runoff, we are done and $I$ is a YES-instance. Otherwise, we might still be able to make $c$ win the runoff by adjusting the weight allocation as follows. We check whether there is a vote $v' \in V_0$ which ranks $c$ higher than $d_1$.

If that is not the case, $I$ is a NO-instance as $c$ can never beat $d_1$ in the runoff. Otherwise, we assign additional weight to $v'$ such that $c$ wins the runoff and,

thus, $I$ is a YES-instance. Note that we do this check before we finally assign the weights, and then assign additional weight at least as high as the sum of the weights of the votes in $V_1$ and $V_0$ with $d_1$ in front of $c$ to $v'$, so that $c$ wins the runoff.

**Case 3:** All candidates in $D_c^{\succ}$ are vetoed by some votes in $V_0$. Hence, we can allocate weights to these votes in $V_0$ such that $c$ beats all candidates in $D_c^{\succ}$. Furthermore, there may be several candidates who can reach the runoff together with $c$, depending on how weights are allocated. Let $e \in C \setminus \{c\}$ be some candidate with the highest score in $V_1$ who is not vetoed by any vote in $V_0$; in case all candidates are vetoed by some vote in $V_0$, we set $e$ to be a candidate with the lowest score in $V_1$. It is clear that a candidate reaching the runoff can only have at most as many vetoes as $e$. Then $D_e^{\succeq} \setminus \{c\}$ is the set of candidates that can potentially join $c$ in the runoff (we achieve this by assigning additional weight to votes in $V_0$ vetoing candidates that score higher than the candidate that we want in the second stage in $V_1$). We now iterate over all candidates $d_1 \in D_e^{\succeq} \setminus \{c\}$ and check whether we can allocate weights such that $c$ wins the runoff against $d_1$:

1. If there is a vote in $V_0$ which does not veto $d_1$ and ranks $c$ in front of $d_1$, we have a YES-instance as we can assign additional weight to this vote (e.g., $|V_1|(|V_0|+1)$) such that $c$ wins the runoff against $d_1$. It is important that this vote does not veto $d_1$, as otherwise, it might happen that $d_1$ does not enter the runoff at all because of the additional weight.
2. If there is no vote in $V_0$ vetoing $d_1$ and ranking $c$ in front of $d_1$, we check if $c$ can win or tie the runoff against $d_1$ by assigning as little weight as possible in order to make them both reach the runoff (i.e., for each candidate with less vetoes in $V_1$ than $d_1$ we assign weight to a vote in $V_0$ vetoing her until she has the same number of vetoes). If this is possible, we have a YES-instance, and otherwise we skip to the next candidate of the iteration.
3. If there are votes in $V_0$ vetoing $d_1$ but no votes in $V_0$ not vetoing $d_1$ and ranking $c$ in front of $d_1$, we assign, similarly to the previous case, as little weight as possible in order to make them both reach the runoff and then allocate as much additional weight as possible to votes in $V_0$ vetoing $d_1$ while ensuring that $d_1$ still reaches the runoff. Then we check if $c$ defeats or ties $d_1$ in the runoff. If this is the case, we have a YES-instance, and otherwise we skip to the next candidate in the iteration.

If $c$ cannot win against any candidate of $D_e^{\succeq} \setminus \{c\}$ in the runoff, we have a NO-instance.

Obviously, all steps can be done in polynomial time, which shows that the problem belongs to P. ❑

To conclude, we have answered all open questions for veto with runoff.

### *5.5. Copeland$^{\alpha}$, Ranked Pairs, Bucklin, and Fallback*

We now study not only a single voting system but instead several voting systems: the family of Copeland$^{\alpha}$ elections for each rational number $\alpha \in [0,1]$, ranked

pairs, Bucklin, and fallback elections. We will prove NP-completeness of all corresponding variants of the possible winner with uncertain weights problem, starting with Copeland$^\alpha$.

**Theorem 5.19.** *For each rational number $\alpha \in [0,1]$, all four variants of* Copeland$^\alpha$-PWUW-$\mathbb{N}$ *are* NP*-complete.*

**Proof.** As for the previous NP-completeness proofs, NP membership is easy to see.

We first prove NP-hardness for Copeland$^\alpha$-PWUW-$\mathbb{N}$ and afterwards explain how to extend the proof to the remaining variants PWUW-BW-$\mathbb{N}$, PWUW-RW-$\mathbb{N}$, and PWUW-BW-RW-$\mathbb{N}$ for Copeland$^\alpha$.

Now, to prove that Copeland$^\alpha$-PWUW-$\mathbb{N}$ is NP-hard, we provide a polynomial-time many-one reduction from X3C. Let $I = (\mathscr{B}, \mathscr{S})$ with $\mathscr{B} = \{b_1, \ldots, b_{3q}\}$ and $\mathscr{S} = \{S_1, \ldots, S_n\}$ be an X3C instance. We construct a Copeland$^\alpha$-PWUW-$\mathbb{N}$ instance $I' = (C, V_1, V_0, c)$ as follows. First, define the set of candidates as $C = \mathscr{B} \cup \{c, d, e\}$, where $c$ is the distinguished candidate. Next, for every $i$, $1 \leq i \leq n$, we add a vote of the form

$$v_i = d > e > \overrightarrow{S_i} > c > \cdots$$

to $V_0$. Finally, to define the votes in $V_1$, we construct the following WMG $G = (C, E)$. (Note that the vertices of $G$ are the candidates of the election.) To define the weighted edge set $E$ to contain

- the edges $(c,d)$, $(d,e)$, and $(e,c)$, each with weight $q+1$, and
- for every $i$, $1 \leq i \leq 3q$,
  - the edges $(d,b_i)$ and $(e,b_i)$, both with weight $q+1$, and
  - the edge $(b_i,c)$, with weight $q-3$.

All other edges not mentioned explicitly are defined to have weight at most 1. The resulting weighted majority graph is illustrated in Figure 3 and we can derive the corresponding votes for $V_1$ by McGarvey's trick as explained in Section 2.

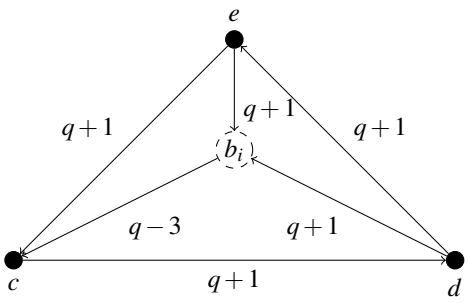



Figure 3: Weighted majority graph $G = (C, E)$ used to provide a polynomial-time many-one reduction from X3C to Copeland$^\alpha$-PWUW-$\mathbb{N}$. We only depict a single vertex $b_i$ representing all vertices from $\mathscr{B}$.

This completes the construction of $I'$, which obviously can be done in time polynomial in $|I|$. Before we now prove that $I \in$ X3C if and only if $I' \in$ Copeland$^\alpha$-PWUW-$\mathbb{N}$, let us make some observations with respect to $I'$. First of all, note that in the pairwise comparison, no matter what weights we allocate to the votes in $V_0$,

- $d$ always beats $e$,
- $e$ always beats $c$, and
- $d$ and $e$ always beat all $b_i$, $1 \leq i \leq 3q$.

For example, that $d$ always beats $e$ is because $d$ appears $q+1$ times more often ahead of $e$ in the votes in $V_1$ than the other way around and all votes in $V_0$ have $d$ ahead of $e$; analogous arguments hold for the other two observations. Furthermore, paying only attention to the votes in $V_1$, $c$ wins against $d$ as $c$ appears $q+1$ times more often ahead of $d$ than $d$ ahead of $c$. Also, every $b_i$, $1 \leq i \leq 3q$, wins against $c$, as every $b_i$ appears $q-3$ times more often ahead of $c$ than $c$ ahead of $b_i$.

Table 3: Copeland$^\alpha$ scores of the candidates in $(C,V_1)$ from instance $I'$.

| | $c$ | $d$ | $e$ | $b_i$ | $\sum$ |
|---|---|---|---|---|---|
| $c$ | – | 1 | 0 | 0 | 1 |
| $d$ | 0 | – | 1 | 1 | $3q+1$ |
| $e$ | 1 | 0 | – | $3q$ | $3q+1$ |
| $b_i$ | 1 | 0 | 0 | $\leq 1$ | $\leq 3q$ |

Table 3 summarizes the resulting Copeland$^\alpha$ scores of the candidates in the election $(C,V_1)$ from instance $I'$. From the previous observations we can conclude that $e$, no matter what weights we allocate to the votes in $V_0$, always has a Copeland$^\alpha$ score of $3q+1$. Furthermore, $d$ always has a Copeland$^\alpha$ score of at least $3q+1$ and at most $3q+2$ (it is $3q+2$ if $d$ also beats $c$), and all $b_i$, $1 \leq i \leq 3q$, have a Copeland$^\alpha$ score of at most $3q$ (it is $3q-1$ if $c$ beats the respective $b_i$). Finally, $c$ can have a Copeland$^\alpha$ score of at most $3q+1$, namely if $c$ beats $d$ and all $b_i$, $1 \leq i \leq 3q$.

Now, let us show that $I \in$ X3C if and only if $I' \in$ Copeland$^\alpha$-PWUW-$\mathbb{N}$.

From left to right, assume that $I$ is a YES-instance X3C. Then there exists an exact set cover $\mathscr{S}' \subseteq \mathscr{S}$ of size $|\mathscr{S}'| = q$. For every $S_i \in \mathscr{S}'$, we allocate weight $w_i = 1$ to $v_i \in V_0$. Consequently, in total, we allocate a weight of $q$ to the votes in $V_0$. Thus $c$ still beats $d$ in the weighted election $(C,V_1 \cup V_0)$, as $c$ had a point advantage of $q+1$ from $V_1$ which now has shrunk to 1. Furthermore, since $\mathscr{S}'$ is an exact cover of $\mathscr{B}$, every $b_i$ appears exactly once ahead of $c$ and $q-1$ times behind $c$ in the votes from $V_0$ with weight 1. Consequently, the point advantage of every $b_i$ over $c$ flipped from $q-3$ to $q-3+1-(q-1) = -1$, i.e., $c$ now wins against all $b_i$, $1 \leq i \leq 3q$. Thus, $e$, $d$, and $c$ obtain a Copeland$^\alpha$ score of $3q+1$ while every $b_i$ obtains a Copeland$^\alpha$ score of at most $3q-1$, so $c$ is a winner of $(C,V_1 \cup V_0)$ and therefore, $I'$ a YES-instance of Copeland$^\alpha$-PWUW-$\mathbb{N}$.

Conversely, from right to left, assume that $I'$ is a YES-instance of Copeland$^\alpha$-PWUW-$\mathbb{N}$. In that case, we have a weight allocation to the votes in $V_0$ such that $c$

wins the weighted election $(C, V_1 \cup V_0)$. From our previous observations we know that $c$ can only win the election if $c$ beats $d$ and all $b_i$, $1 \leq i \leq 3q$. Thus, the sum of all weights allocated to votes in $V_0$ must be at most $q$, as otherwise $c$ no longer beats $d$. If the sum of all weights is less than $q$, there is at least one candidate $b_i$, that appears once ahead of $c$ in the positively weighted votes from $V_0$ but at most $q-2$ times behind $c$, so $c$ does not win against $b_i$, causing $c$ to not be a winner of the weighted election. Therefore, the only possibility is that the overall weight allocated to the votes in $V_0$ equals $q$, and every candidate from $\mathscr{B}$ appears exactly once ahead of $c$, yielding an exact set cover via the positively weighted votes in $V_0$, so $I$ is a YES-instance of X3C. It follows that Copeland$^\alpha$-PWUW-$\mathbb{N}$ is NP-complete.

To see that Copeland$^\alpha$-PWUW-BW-$\mathbb{N}$ is NP-complete, we can add a bound of $B = q$ to $I'$. For Copeland$^\alpha$-PWUW-RW-$\mathbb{N}$, we define the range of every vote in $V_0$ as $\{0,1\}$, and we combine both restrictions for Copeland$^\alpha$-PWUW-BW-RW-$\mathbb{N}$. ❑

Faliszewski et al. [32] proved that Copeland$^\alpha$-CCAV is NP-complete for all rationals $\alpha$, $0 \leq \alpha \leq 1$. Combining this result with Proposition 3.2 offers an alternative proof for the NP-completeness of Copeland$^\alpha$-PWUW-BW-RW-$\mathbb{N}$ just proven. Note, however, that Theorem 5.19 establishes NP-completeness for more problems than just Copeland$^\alpha$-PWUW-BW-RW-$\mathbb{N}$.

Having settled the complexity of all four problem variants for the voting rule Copeland$^\alpha$, we now turn to ranked pairs to show the same results.

**Theorem 5.20.** *All four problem variants of* ranked-pairs-PWUW-$\mathbb{N}$ *are* NP-*complete.*

**Proof.** NP membership is again easy to see for all four variants.

We first prove NP-hardness for ranked-pairs-PWUW-$\mathbb{N}$ and then explain how to extend it to the remaining three variants. To show NP-hardness for ranked-pairs-PWUW-$\mathbb{N}$, we provide a polynomial-time many-one reduction from X3C. Let $I = (\mathscr{B}, \mathscr{S})$ with $\mathscr{B} = \{b_1, \ldots, b_{3q}\}$ and $\mathscr{S} = \{S_1, \ldots, S_n\}$ be some given X3C instance. We construct the ranked-pairs-PWUW-$\mathbb{N}$ instance $I' = (C, V_1, V_0, c)$ as follows. First, define the set of candidates as $C = \mathscr{B} \cup \{c, d, e\}$, where $c$ is the distinguished candidate. Next, the set $V_0$ of votes without assigned weights consists of

$$v_i = e > \overrightarrow{S_i} > c > d > \cdots$$

for $1 \leq i \leq n$. Finally, the votes in $V_1$ are derived from the following WMG $G = (C, E)$ via McGarvey's trick, where again the vertices of $G$ are the candidates of the election. We specify the the following weighted edges in $E$:

- $(c,d)$ and $(e,c)$ with weight $2q+1$,
- $(d,e)$ with weight $4q+1$, and
- for every $i$, $1 \leq i \leq 3q$,
  - the edges $(d,b_i)$ and $(e,b_i)$, both with weight $2q+1$, and
  - the edge $(b_i,c)$ with weight $4q-1$.

All other edges not explicitly mentioned are defined to have a weight of at most 1. The resulting graph $G$ looks similarly to the WMG in Figure 3, except with different weights assigned to the edges. This finishes the construction of $I'$ in time polynomial in $|I|$. Table 4 shows the pairwise score differences for the votes in $V_1$.

Table 4: Pairwise score differences for candidates in $C$ based on the votes in $V_1$ from instance $I'$.

| | $c$ | $d$ | $e$ | $b_i$ |
|---|---|---|---|---|
| $c$ | – | $2q+1$ | $-(2q+1)$ | $-(4q-1)$ |
| $d$ | $-(2q+1)$ | – | $4q+1$ | $2q+1$ |
| $e$ | $2q+1$ | $-(4q+1)$ | – | $2q+1$ |
| $b_i$ | $4q-1$ | $-(2q+1)$ | $-(2q+1)$ | $\leq 1$ |

Next, we prove that $I$ is a YES-instance of X3C if and only if $I'$ is a YES-instance of ranked-pairs-PWUW-$\mathbb{N}$.

From left to right, assume that $I$ is a YES-instance of X3C. Then there exists an exact set cover $\mathscr{S}' \subseteq \mathscr{S}$ with $|\mathscr{S}'| = q$. For every $S_i \in \mathscr{S}'$, we set the weight of vote $v_i \in V_0$ to 1, and for all other votes in $V_0$ to 0. Consequently, every $b_i$, $1 \leq i \leq 3q$, appears exactly once ahead of $c$ and $d$ and $q-1$ times behind $c$ and $d$. Table 5 shows the resulting score differences. From this table we can see that there is a tie between all candidates such that, assuming a tie-breaking mechanism in favor of $c$, $c$ is a winner of the weighted election $(C, V_1 \cup V_0)$ and hence, $I'$ is a YES-instance of ranked-pairs-PWUW-$\mathbb{N}$.

Table 5: Pairwise score differences for candidates in $C$ based on the votes in $V_1 \cup V_0$ from instance $I'$.

| | $c$ | $d$ | $e$ | $b_i$ |
|---|---|---|---|---|
| $c$ | – | $3q+1$ | $-(3q+1)$ | $-(3q+1)$ |
| $d$ | $-(3q+1)$ | – | $3q+1$ | $3q-1$ |
| $e$ | $3q+1$ | $-(3q+1)$ | – | $3q+1$ |
| $b_i$ | $3q+1$ | $-(3q-1)$ | $-(3q+1)$ | $\leq q+1$ |

From right to left, assume that $I'$ is a YES-instance of ranked-pairs-PWUW-$\mathbb{N}$. Consequently, we have a weight allocation to the votes in $V_0$ such that $c$ wins the weighted election $(C, V_1 \cup V_0)$. We distinguish the following three cases:

**Case 1:** $\sum_{v_i \in V_0} w_i > q$. In this case, it follows from the way how the votes in $V_0$ are constructed that $(e,c), (c,d)$, and $(e,b_i)$, $1 \leq i \leq 3q$, have a weight of at least $3q+2$ while $(d,e)$ has a weight of at most $3q$. Thus, no matter what the weights of $(c,b_i), (d,b_i)$, and $(b_i,b_j)$ are, the preference relations $e > c$, $e > b_i$, and $c > d$ are always fixed first, enforcing $e > d$ due to transitivity, before $(d,e)$ is considered. It follows that $e$ is preferred to each of $c$, $d$, and $b_i$; hence, $c$ cannot be a winner of the weighted election $(C, V_1 \cup V_0)$, a contradiction to our initial assumption, so this case is impossible.

**Case 2:** $\sum_{v_i \in V_0} w_i < q$. If the sum of all weights of voters in $V_0$, denoted by $\tau = \sum_{v_i \in V_0} w_i$, satisfies $\tau < q$, then we have a weight of $4q+1-\tau$ for $(d,e)$, a weight of $2q+1+\tau$ for $(e,c)$, and a weight of $2q+1+\tau$ for $(c,d)$. Additionally, there must be at least one $b_i^*$ who appears at least once in front of $c$ and $d$ and only $\tau-1$ times behind $c$ and $d$, so we have a weight of $4q-1+1-(\tau-1) = 4q+1-\tau$ for $(b_i^*,c)$. With $\tau < q$ we obtain that

$$4q+1-\tau > 2q+1+\tau.$$

Therefore, the final order contains $b_i^* > c > d > e$, so $c$ cannot win the weighted election $(C, V_1 \cup V_0)$, again a contradiction to our initial assumption making this case impossible.

**Case 3:** $\sum_{v_i \in V_0} w_i = q$. From the previous two cases we know that this third case is the only possible case. With the sum of the weights allocated to the votes in $V_0$ being equal to $q$, it follows that the three edges $(c,d)$, $(d,e)$, and $(e,c)$ each have a weight of $3q+1$. If every candidate in $C \setminus \{c,d,e\} = \mathscr{B}$ appears exactly once ahead of $c$ and $d$ and $q-1$ times behind $c$ and $d$, it follows that $(b_i,c)$ has a weight of $3q+1$ and $(d,b_i)$ has a weight of $3q-1$, so there is a tie among $c$, $d$, $b_i$, and $e$ (which means that $c$ wins the weighted election $(C, V_1 \cup V_0)$ with a tie break in favor of $c$). If, however, there is at least one candidate $b_i$ appearing at least twice in front of $c$ and $d$ and hence at most $q-2$ times behind $c$ and $d$, it follows that the edge $(b_i,c)$ would have a weight of $4q-1+2-(q-2) = 3q+3 > 3q+1$, so $b_i$ would win the election, contradicting our assumption that $c$ wins the weighted election. It follows that every candidate $b_i$ appears exactly once in front of $c$ and $d$ and $q-1$ times behind these two candidates. Therefore, the positively weighted votes from $V_0$, each with weight 1, form an exact set cover for $\mathscr{B}$ of size $q$.

Thus ranked-pairs-PWUW-$\mathbb{N}$ is NP-complete. To extend this proof to ranked-pairs-PWUW-BW-$\mathbb{N}$, we add a bound $B = q$ to $I'$. For ranked-pairs-PWUW-RW-$\mathbb{N}$, we add regions $\{0,1\}$ for all votes in $V_0$ to $I'$, and we combine both restrictions for ranked-pairs-PWUW-BW-RW-$\mathbb{N}$. ❑

Next, we consider Bucklin voting.

**Theorem 5.21.** *All four problem variants of* Bucklin-PWUW-$\mathbb{N}$ *studied here are* NP-*complete.*

**Proof.** As for the previous NP-completeness results, NP membership is easy to see.

To prove NP-hardness, we first show that Bucklin-PWUW-$\mathbb{N}$ is NP-hard and afterwards extend the proof to the remaining variants.

The NP-hardness result for Bucklin-PWUW-$\mathbb{N}$ follows by a polynomial-time many-one reduction from X3C. Let $I = (\mathscr{B}, \mathscr{S})$ with $\mathscr{B} = \{b_1, \ldots, b_{3q}\}$ and $\mathscr{S} = \{S_1, \ldots, S_n\}$ be some given X3C instance. We construct a Bucklin-PWUW-$\mathbb{N}$ instance $I' = (C, V_1, V_0, c)$ as follows. First, define the set of candidates as $C = \mathscr{B} \cup \{c,d\} \cup D \cup$

$D'$ with $D = \{d_1, \ldots, d_{3q}\}$ and $D' = \{d'_1, \ldots, d'_{3q}\}$ as auxiliary candidates and $c$ as the distinguished candidate. Next, for every $S_i \in \mathscr{S}$, we add a vote of the form

$$v_i = d > \overrightarrow{S_i} > c > \overrightarrow{D} > \overrightarrow{D'} > \mathscr{B} \setminus S_i$$

to $V_0$. Lastly, we add $q-1$ copies of the vote $v = \overrightarrow{\mathscr{B}} > c > \overrightarrow{D'} > \overrightarrow{D} > d$ and one vote of the form $\overrightarrow{D'} > c > \overrightarrow{\mathscr{B}} > d > \overrightarrow{D}$ to $V_1$. This completes the construction of $I'$ in time polynomial in $|I|$.

Now, we prove that $I$ is a YES-instance of X3C if and only if $I'$ is a YES-instance of Bucklin-PWUW-$\mathbb{N}$.

From left to right, assume that $I$ is a YES-instance of X3C. Then there exists an exact set cover $\mathscr{S}' \subseteq \mathscr{S}$ of size $|\mathscr{S}'| = q$ for $\mathscr{B}$. For every $S_i \in \mathscr{S}'$, we set the weight of the corresponding vote $v_i \in V_0$ to 1, and to 0 for all remaining votes in $V_0$. Doing so, we obtain a total sum of $2q$ for the votes in $V_1 \cup V_0$. Table 6 lists the corresponding Bucklin scores of all candidates in $C$.

Table 6: Bucklin scores for the candidates in $C$ if $I$ is a YES-instance of X3C.

| $c$ | $d$ | $b_i$ | $d_i$ | $d'_i$ |
|---|---|---|---|---|
| $3q+1$ | $6q+2$ | $\geq 3q+2$ | $\geq 6q+2$ | $\geq 3q+6$ |

One can see that $c$ has the smallest Bucklin score and, hence, wins the weighted election $(C, V_1 \cup V_0)$, so $I'$ is a YES-instance of Bucklin-PWUW-$\mathbb{N}$.

From right to left, assume that $I'$ is a YES-instance of Bucklin-PWUW-$\mathbb{N}$. In this case, we have some weights $w_i \in \mathbb{N}$ for the votes $v_i \in V_0$ such that $c$ wins the weighted election $(C, V_1 \cup V_0)$. If the sum of the weights of the votes in $V_0$ is greater than $q$ (i.e., if $\sum_{v_i \in V_0} w_i > q$), $d$ obviously has a Bucklin score of 1, whereas $c$ has a Bucklin score greater than 1, so $d$ wins the weighted election, a contradiction to our initial assumption that $c$ wins. If the sum of weights of the votes in $V_0$ is less than $q$ (i.e., if $\sum_{v_i \in V_0} w_i < q$), then there is at least one $b_i \in \mathscr{B}$ that appears at least once at the top of one of the positively weighted votes from $V_0$ and $q-1$ times in the votes from $V_1$, so this candidate has a Bucklin score of at most $3q$, whereas $c$ has a Bucklin score of at least $3q+1$, again a contradiction to our initial assumption that $c$ wins the election. Consequently, $\sum_{v_i \in V_0} w_i = q$ must hold. Now, if there is one $b_i$ that appears at the top of more than one positively weighted vote from $V_0$, this candidate $b_i$ has a Bucklin score of at most $3q$ while $c$ has a Bucklin score of $3q+1$, which would again contradict our assumption that $c$ wins the election. Hence, every candidate $b_i$ from $\mathscr{B}$ must appear exactly once at the top of one positively weighted vote from $V_0$, so these votes form an exact set cover for $\mathscr{B}$ of size $q$, yielding that $I$ is a YES-instance of X3C.

This shows that Bucklin-PWUW-$\mathbb{N}$ is NP-complete. To see that Bucklin-PWUW-BW-$\mathbb{N}$ is NP-complete, we can add a bound of $B = q$ to the instance $I'$. To see that Bucklin-PWUW-RW-$\mathbb{N}$ is NP-complete, we can add regions $\{0,1\}$ for all votes in $V_0$ to $I'$, and to see that Bucklin-PWUW-BW-RW-$\mathbb{N}$ is NP-complete, we combine both of the previous requirements for $I'$. ❑

Bucklin voting can be seen as the special case of fallback voting where all votes consist of complete linear orders over the candidates. Hence, the NP-hardness results for Bucklin voting can immediately be transferred to fallback voting. NP membership for fallback follows by the same arguments as for the previous voting rules. We thus have the following corollary.

**Corollary 5.22.** *All four problem variants of* fallback-PWUW-$\mathbb{N}$ *are* NP*-complete.*

*5.6. Borda*

Finally, we turn to the voting rule due to Borda [16], which perhaps is the most famous scoring protocol and has been intensively studied in social choice theory and in computational social choice (see the survey by Rothe [56] and the discussion of related work by Neveling and Rothe [51]). As for $k$-veto, $k \geq 3$, we establish hardness results for Borda, but now even for all four of our problem variants.

**Theorem 5.23.** *For* Borda*, all four problem variants of* PWUW-$\mathbb{N}$ *are* NP*-complete.*

**Proof.** Membership in NP is obvious for all problem variants but Borda-PWUW-$\mathbb{N}$. For Borda-PWUW-$\mathbb{N}$, it is not trivial to show that a solution that can be used as a witness is polynomial in the input size. But in this case we can construct in polynomial time an integer linear program that solves the problem similarly to the linear programs that we constructed in Section 4 for the variants with rational weights. Then NP membership of Borda-PWUW-$\mathbb{N}$ follows from the reduction from Borda-PWUW-$\mathbb{N}$ to the NP-complete problem INTEGER-PROGRAMMING-FEASIBILITY.[8]

Regarding the four NP-hardness results, we will prove NP-hardness for Borda-PWUW-$\mathbb{N}$ and then describe how the reduction can be extended to obtain NP-hardness for the remaining cases. So, in order to prove that Borda-PWUW-$\mathbb{N}$ is NP-hard, we again reduce from X3C. Let $I = (\mathscr{B}, \mathscr{S})$ be a given X3C instance with $\mathscr{B} = \{b_1, \ldots, b_{3q}\}$ and $\mathscr{S} = \{S_1, \ldots, S_n\}$. We construct a Borda-PWUW-$\mathbb{N}$ instance $I' = (C, V_1, V_0, c)$ as follows. Define $C = \{c, d, d'\} \cup \mathscr{B} \cup X$ with $X$ being a set of $9q^2$ buffer candidates. For each $b_i \in \mathscr{B}$ and $x_j \in X$, we add the two votes

$$\overrightarrow{C \setminus \{b_i, x_j\}} > b_i > x_j \quad \text{and} \quad b_i > x_j > \overleftarrow{C \setminus \{b_i, x_j\}}$$

and $q$ times two votes of the form

$$\overrightarrow{C \setminus \{d, d'\}} > d' > d \quad \text{and} \quad d' > d > \overleftarrow{C \setminus \{d, d'\}}$$

to $V_1$. Doing so, we construct the following point distances between $c$ and each of the other candidates in the election $(C, V_1)$:

1. $\mathit{diff}_{(C,V_1)}(c, d) = q$,
2. $\mathit{diff}_{(C,V_1)}(c, d') = -q$,

[8] We note in passing that with this technique we can show NP membership of $\mathscr{E}$-PWUW-$\mathbb{N}$ for every scoring protocol $\mathscr{E}$.

3. $\mathit{diff}_{(C,V_1)}(c,b_i) = -|X|$ for every $b_i \in \mathscr{B}$, and
4. $\mathit{diff}_{(C,V_1)}(c,x_j) = 3q$ for every $x_j \in X$.

For each $i$, $1 \leq i \leq n$, we add a vote of the form

$$v_i = d > c > \mathscr{B} \setminus S_i > X > S_i > d'$$

to $V_0$. Obviously, this construction of $I'$ is possible in time polynomial in $|I|$.

Note that $c$ does not win in $(C,V_1)$, and no matter how many votes from $V_0$ we weigh positively, we know that

- $c$ beats $d'$, as $c$ gains at least $q+1$ points more than $d'$ from each vote in $V_0$; and
- $c$ beats every buffer candidate $x_j \in X$, as $c$ is in front of each of them in all votes of $V_0$ and already beats them in $(C,V_1)$.

Thus we only need to worry about the point balances between $c$ and $d$ and between $c$ and each candidate in $\mathscr{B}$.

It remains to prove that $I \in$ X3C if and only if $I' \in$ Borda-PWUW-$\mathbb{N}$.

From left to right, assume that $I \in$ X3C. Hence, there is an exact cover $\mathscr{S}' \subseteq \mathscr{S}$ with $|\mathscr{S}'| = q$ such that $\bigcup_{S_i \in \mathscr{S}'} S_i = \mathscr{B}$. For every $S_i \in \mathscr{S}'$, we set the weight of the corresponding vote $v_i \in V_0$ to $w_i = 1$ and for all remaining votes in $V_0$ to 0. Since we have assigned a total weight of $q$ to the votes in $V_0$ and $c$ is directly behind $d$ in all those votes, $c$ now ties with $d$. The votes with weight 1 correspond to an exact cover, so for each $b_i \in \mathscr{B}$ with $b_i \in S_j$ such that $S_j \in \mathscr{S}'$, $c$ gains at least $|X|$ points on $b_i$ from the vote $v_j$ and is in front of $b_i$ in all other votes from $V_0$. Thus $c$ beats or ties all candidates of $\mathscr{B}$. Therefore, $c$ is a winner of the weighted election $(C,V_1 \cup V_0)$, which means that $I' \in$ Borda-PWUW-$\mathbb{N}$.

From right to left, assume that $I \notin$ X3C. Then, for every $\mathscr{S}' \subseteq \mathscr{S}$ with $|\mathscr{S}'| \leq q$, there exists at least one $b_i \in \mathscr{B}$ such that $b_i \notin \bigcup_{S_j \in \mathscr{S}'} S_j$. First, note that we cannot assign a total weight of more than $q$ to the votes in $V_0$ since, for every vote of positive weight, $c$ loses at least one point on $d$, as $c$ is directly behind $d$ in all votes of $V_0$ and has a point buffer of only $q$ from the votes in $V_1$. Now, assume a total weight of at most $q$ has been assigned to the votes in $V_0$ and let $\mathscr{S}' \subseteq \mathscr{S}$ be such that for each $S_j \in \mathscr{S}'$ the corresponding vote $v_j$ has weight one or more. Obviously, $|\mathscr{S}'| \leq q$. Then $\mathscr{S}'$ cannot be a cover of $\mathscr{B}$, i.e., there is some $b_i \in \mathscr{B}$ that is not covered by $\mathscr{S}'$ (formally, $b_i \notin \bigcup_{S_j \in \mathscr{S}'} S_j$). Therefore, for each $v_j$ such that $S_j \in \mathscr{S}'$ and $b_i \in \mathscr{B} \setminus S_j$, $c$ only gains at most $q-3$ points more than $b_i$ from $v_j$, which sums up to at most $q(q-3) = q^2 - 3q$ points that $c$ gains more than $b_i$ from the votes in $V_0$. This is not enough, however, to make up for the negative point balance of $-|X| = -9q^2$ to $b_i$ from the votes in $V_1$, so $c$ is beaten by $b_i$ and cannot win if only a total weight of at most $q$ can be assigned to votes in $V_0$. It follows that $I' \notin$ Borda-PWUW-$\mathbb{N}$.

This concludes the reduction and the proof of NP-hardness for Borda-PWUW-$\mathbb{N}$.

To show NP-hardness for Borda-PWUW-RW-$\mathbb{N}$ and Borda-PWUW-BW-$\mathbb{N}$, we augment the above constructed instance $I'$ with a budget of $q$ to obtain the instance $I_{BW} = (C,V_1,V_0,q,c)$ or with a set of regions $R = \{R_1,\ldots,R_{|V_0|}\}$ with $R_i = \{0,1\}$ for all $v_i \in V_0$, to obtain $I_{RW} = (C,V_1,V_0,R,c)$. In the proof of correctness above, we never allocate a total weight of more than $q$ and all allocated weights are either 0 or 1. This

Table 7: Overview of complexity results for nonnegative rational weights from $\mathbb{Q}^+$.

| | Plurality with runoff | Veto with runoff | Scoring Rules | Bucklin, Fallback |
|---|---|---|---|---|
| $\mathbb{N}$ | P | P | P | P |
| BW-RW-$\mathbb{Q}^+$ | P | P | P | P |
| BW-$\mathbb{Q}^+$ | P | P | P | P |
| RW-$\mathbb{Q}^+$ | P | P | P | P |

Table 8: Overview of complexity results for nonnegative integer weights. "NP-c." stands for NP-complete and "$k$-AV" stands for $k$-approval.

| | $k$-AV, $k \leq 3$ | $k$-AV, $k \geq 4$ | $k$-veto, $k \leq 2$ | $k$-veto, $k \geq 3$ | Plurality/Veto with runoff | Borda | Bucklin, Fallback | Copeland, Ranked Pairs |
|---|---|---|---|---|---|---|---|---|
| $\mathbb{N}$ | P | P | P | P | P | NP-c. | NP-c. | NP-c. |
| BW-RW-$\mathbb{N}$ | P | NP-c. | P | NP-c. | P | NP-c. | NP-c. | NP-c. |
| BW-$\mathbb{N}$ | P | NP-c. | P | NP-c. | P | NP-c. | NP-c. | NP-c. |
| RW-$\mathbb{N}$ | P | P | P | P | P | NP-c. | NP-c. | NP-c. |

directly yields $I \in$ X3C $\Leftrightarrow I_{BW} \in$ Borda-PWUW-BW-$\mathbb{N}$ and $I \in$ X3C $\Leftrightarrow I_{RW} \in$ Borda-PWUW-RW-$\mathbb{N}$. NP-hardness of Borda-PWUW-BW-RW-$\mathbb{N}$ is inherited from Borda-PWUW-BW-$\mathbb{N}$ or Borda-PWUW-RW-$\mathbb{N}$. ❑

With this result we finish our study of the possible winner with uncertain weights problem for nonnegative integer weights.

## 6. Conclusion

We introduced the possible winner with uncertain weights problem in which not the preferences but the weights of the votes are uncertain, and we studied this problem and its variants in a general framework. We showed that some of these problem variants are easy and some are hard to solve for nine of the most important voting systems. We studied them for nonnegative rational weights and for nonnegative integer weights. Table 7 provides an overview of the former results and Table 8 an overview of the latter results.

Interestingly, while the original possible winner problem (in which there is uncertainty about the voters' preferences) generalizes the coalitional manipulation problem and is a special case of swap bribery [26], the possible winner with uncertain weights problem generalizes the problem of constructive control by adding or deleting voters, as pointed out in Section 3. In the course of our research, we have also further completed the landscape of results with respect to CCAV *in succinct representation* by showing that this problem can be solved in polynomial time for both plurality with runoff and veto with runoff.

As to open problems, like for the *possible winner problem* studied by Betzler and Dorn [14] and Baumeister and Rothe [3], it is desirable to have a dichotomy result with respect to *all* scoring protocols for our problem variants with nonnegative integer weights.

Another interesting task for future research is to study the *necessary winner with uncertain weights problem*. For this problem our goal is to check whether the distinguished candidate $c$ is a winner for every allowed allocation of weights. In some sense, this can also be seen as the *destructive* variant of the possible winner with uncertain weights problem where the question is whether $c$'s victory can be prevented by some weight assignment. Additionally, it would be interesting to study an even more general variant: the weighted possible winner problem with uncertainty about both the voters' preferences and their weights.

Finally, one could also define other variants of $\mathscr{E}$-PWUW-BW-RW-$\mathbb{F}$ and $\mathscr{E}$-PWUW-RW-$\mathbb{F}$, e.g., by allowing *sets of intervals* for each weight.

**Acknowledgments**

We thank the anonymous ECAI-2012 and FCT-2021 reviewers for the very helpful comments, and the FCT-2021 program committee for honoring our work [52] with the best paper award. This work was supported in part by an SFF grant of HHU Düsseldorf, DFG grants RO 1202/{14-2, 15-1, 21-1}, ARC-DP110101792, a DAAD-PPP/PROCOPE grant, and by NSF grant #1136996 to the Computing Research Association for the CIFellows Project.

**References**

[1] 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, July 2010.

[2] J. Bartholdi III, C. Tovey, and M. Trick. How hard is it to control an election? *Mathematical and Computer Modelling*, 16(8/9):27–40, 1992.

[3] 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.

[4] 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, May 2011.

[5] D. Baumeister, P. Faliszewski, J. Lang, and J. Rothe. Campaigns for lazy voters: Truncated ballots. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*, pages 577–584. IFAAMAS, June 2012.

[6] 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. IOS Press, August 2012.

[7] D. Baumeister, G. Erdélyi, O. Erdélyi, and J. Rothe. Complexity of manipulation and bribery in judgment aggregation for uniform premise-based quota rules. *Mathematical Social Sciences*, 76:19–30, 2015.

[8] D. Baumeister, S. Bouveret, J. Lang, N. Nguyen, T. Nguyen, J. Rothe, and A. Saffidine. Positional scoring-based allocation of indivisible goods. *Journal of Autonomous Agents and Multi-Agent Systems*, 31(3):628–655, 2017.

[9] D. Baumeister, D. Neugebauer, J. Rothe, and H. Schadrack. Verification in incomplete argumentation frameworks. *Artificial Intelligence*, 264:1–26, 2018.

[10] D. Baumeister, G. Erdélyi, O. Erdélyi, J. Rothe, and A. Selker. Complexity of control in judgment aggregation for uniform premise-based quota rules. *Journal of Computer and System Sciences*, 112:13–33, 2020.

[11] D. Baumeister, M. Järvisalo, D. Neugebauer, A. Niskanen, and J. Rothe. Acceptance in incomplete argumentation frameworks. *Artificial Intelligence*, 295: 103470, 2021.

[12] Dorothea Baumeister and Tobias Hogrebe. How hard is the manipulative design of scoring systems? In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 74–80, 2019.

[13] N. Betzler. On problem kernels for possible winner determination under the $k$-approval protocol. In *Proceedings of the 35th International Symposium on Mathematical Foundations of Computer Science*, pages 114–125. Springer-Verlag *Lecture Notes in Computer Science #6281*, August 2010.

[14] N. Betzler and B. Dorn. Towards a dichotomy for the possible winner problem in elections based on scoring rules. *Journal of Computer and System Sciences*, 76 (8):812–836, 2010.

[15] N. Betzler, S. Hemmann, and R. Niedermeier. A multivariate complexity analysis of determining possible winners given incomplete votes. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pages 53–58. IJCAI, July 2009.

[16] J.-C. de Borda. Mémoire sur les élections au scrutin. *Histoire de L'Académie Royale des Sciences, Paris*, 1781. English translation appears in the paper by de Grazia [23].

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

[18] J. Chamberlin and M. Cohen. A linear inequality method of establishing certain social choice conjectures. *Public Choice*, 33(2):5–16, 1978.

[19] 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. AAAI Press, July 2010.

[20] Y. Chevaleyre, J. Lang, N. Maudet, J. Monnot, and L. Xia. New candidates welcome! Possible winners with respect to the addition of new candidates. *Mathematical Social Sciences*, 64(1):74–88, 2012.

[21] V. Conitzer, T. Sandholm, and J. Lang. When are elections with few candidates hard to manipulate? *Journal of the ACM*, 54(3):Article 14, 2007.

[22] A. Copeland. A "reasonable" social welfare function. Mimeographed notes from a Seminar on Applications of Mathematics to the Social Sciences, University of Michigan, 1951.

[23] A. de Grazia. Mathematical deviation of an election system. *Isis*, 44(1–2):41–51, 1953.

[24] B. Dorn and I. Schlotter. Multivariate complexity analysis of swap bribery. *Algorithmica*, 64(1):126–151, 2012.

[25] 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, June 2012.

[26] 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*, October 2009.

[27] E. Elkind, P. Faliszewski, and A. Slinko. Cloning in elections: Finding the possible winners. *Journal of Artificial Intelligence Research*, 42:529–573, 2011.

[28] G. Erdélyi, M. Fellows, J. Rothe, and L. Schend. Control complexity in Bucklin and fallback voting: An experimental analysis. *Journal of Computer and System Sciences*, 81(4):661–670, 2015.

[29] G. Erdélyi, M. Fellows, J. Rothe, and L. Schend. Control complexity in Bucklin and fallback voting: A theoretical analysis. *Journal of Computer and System Sciences*, 81(4):632–660, 2015.

[30] G. Erdélyi, M. Neveling, C. Reger, J. Rothe, Y. Yang, and R. Zorn. Towards completing the puzzle: Complexity of control by replacing, adding, and deleting candidates or voters. *Journal of Autonomous Agents and Multi-Agent Systems*, 35 (2):41, 2021.

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

[32] 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, 2009.

[33] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. Using complexity to protect elections. *Communications of the ACM*, 53(11):74–82, 2010.

[34] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. Multimode control attacks on elections. *Journal of Artificial Intelligence Research*, 40:305–351, 2011.

[35] Z. Fitzsimmons and E. Hemaspaandra. High-multiplicity election problems. *Autonomous Agents and Multi-Agent Systems*, 33(4):383–402, 2019.

[36] D. Fulkerson and G. Dantzig. Computation of maximal flows in networks. Technical report, RAND CORP SANTA MONICA CA, 1955.

[37] H. Gabow. An efficient reduction technique for degree-constrained subgraph and bidirected network flow problems. In *Proceedings of the 15th ACM Symposium on Theory of Computing*, pages 448–456. ACM Press, 1983.

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

[39] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*. Springer, 1988.

[40] L. Hačijan. A polynomial algorithm in linear programming. *Soviet Mathematics Doklady*, 20(1):191–194, 1979.

[41] E. Hemaspaandra, H. Spakowski, and J. Vogel. The complexity of Kemeny elections. *Theoretical Computer Science*, 349(3):382–391, 2005.

[42] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Anyone but him: The complexity of precluding an alternative. *Artificial Intelligence*, 171(5–6):255–285, 2007.

[43] E. Hemaspaandra, L. Hemaspaandra, and H. Schnoor. A control dichotomy for pure scoring rules. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, pages 712–720. AAAI Press, July 2014.

[44] A. Kerkmann, J. Lang, A. Rey, J. Rothe, H. Schadrack, and L. Schend. Hedonic games with ordinal preferences and thresholds. *Journal of Artificial Intelligence Research*, 67:705–756, 2020.

[45] K. Konczak and J. Lang. Voting procedures with incomplete preferences. In *Proceedings of the Multidisciplinary IJCAI-05 Workshop on Advances in Preference Handling*, pages 124–129, July/August 2005.

[46] B. Kuckuck and J. Rothe. Duplication monotonicity in the allocation of indivisible goods. *AI Communications*, 32(4):253–270, 2019.

[47] 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. ijcai.org, July 2020.

[48] H. Lenstra Jr. Integer programming with a fixed number of variables. *Mathematics of Operations Research*, 8(4):538–548, 1983.

[49] A. Lin. *Solving Hard Problems in Election Systems*. PhD thesis, Rochester Institute of Technology, Rochester, NY, USA, March 2012.

[50] D. McGarvey. A theorem on the construction of voting paradoxes. *Econometrica*, 21(4):608–610, 1953.

[51] M. Neveling and J. Rothe. Control complexity in Borda elections: Solving all open cases of offline control and some cases of online control. *Artificial Intelligence*, 298:103508, 2021.

[52] M. Neveling, J. Rothe, and R. Weishaupt. The possible winner problem with uncertain weights revisited. In *Proceedings of the 23rd International Symposium on Fundamentals of Computation Theory*. Springer-Verlang LNCS, September 2021. To appear.

[53] A. Niskanen, D. Neugebauer, M. Järvisalo, and J. Rothe. Deciding acceptance in incomplete argumentation frameworks. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, pages 2942–2949. AAAI Press, February 2020.

[54] C. Papadimitriou. *Computational Complexity*. Addison-Wesley, second edition, 1995.

[55] J. Rothe. *Complexity Theory and Cryptology. An Introduction to Cryptocomplexity*. EATCS Texts in Theoretical Computer Science. Springer-Verlag, 2005.

[56] J. Rothe. Borda count in collective decision making: A summary of recent results. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, pages 9830–9836. AAAI Press, January/February 2019.

[57] K. Skiba, D. Neugebauer, and J. Rothe. Complexity of nonempty existence problems in incomplete argumentation frameworks. *IEEE Intelligent Systems*, 36(2): 13–24, 2021.

[58] L. Xia and V. Conitzer. Determining possible and necessary winners given partial orders. *Journal of Artificial Intelligence Research*, 41:25–67, 2011.

[59] L. Xia, J. Lang, and J. Monnot. Possible winners when new alternatives join: New results coming up! In *Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems*, pages 829–836. IFAAMAS, May 2011.

# 5 Cutting a Cake Is Not Always a "Piece of Cake": A Closer Look at the Foundations of Cake-Cutting Through the Lens of Measure Theory

## 5.1 Summary

In this work we studied the axiomatic foundations of cake-cutting. In particular, we were interested in an optimal definition of $\mathcal{P} \subseteq \mathfrak{P}(X)$, the set of all admissible pieces of cake containing all pieces of cake that could potentially be allocated during the execution of a cake-cutting protocol. First, we surveyed the existing cake-cutting literature and found that across the literature several different approaches exist with respect to defining the set of all admissible pieces of cake. The approaches we found range from defining $\mathcal{P}$ as the set containing all finite unions of subintervals from $X$ up to $\mathcal{P}$ equaling to the power set of $X$, i.e., $\mathfrak{P}(X)$. Based on the generally accepted requirements for the set of all admissible pieces of cake in the cake-cutting literature, we determined that the set of all admissible pieces of cake must form an algebra over the cake $X$ itself. Since the set of all admissible pieces of cake is the domain of every valuation function, we were able to combine our previous finding with the generally accepted expectations for valuation functions and formalized requirements that every agent's valuation function must satisfy. Doing so, we deduced that every valuation function must be a finite content on the set of all admissible pieces of cake.

Equipped with these two new formal definitions for the set of all admissible pieces of cake and valuation functions, we investigated what would be the most preferred choice for the set of all admissible pieces of cake among the choices used in the existing literature. The overall goal was to find a definition for $\mathcal{P}$ that is as large as possible in order to allow as many as possible different pieces of cake to be cut during the execution of a protocol and prevent unnecessary limitations. Giving a mathematically advanced example based on Vitali sets, we showed that the common concept of box-based valuation functions is not powerful enough to satisfy the power set over $X$ as choice for the set of all admissible pieces of cake. Furthermore, we constructed a finite content over $\mathfrak{P}(X)$, based on Banach limits, that would satisfy all requirements of a valuation function and discussed its shortcomings, e.g., its reliance on the validity of the axiom of choice. Both observations, the example and the constructed valuation function, explained why $\mathfrak{P}(X)$ is not a well suited choice for the set of all admissible pieces of cake.

We advanced the problem from a measure-theoretic point of view and argued why the Borel $\sigma$-algebra over $X$, denoted by $\mathcal{B}(X)$, is our recommended choice for the set of all admissible pieces of cake. Not only is the Borel $\sigma$-algebra a well-studied and thoroughly understood notion from measure theory, but if one decides to use the Borel $\sigma$-algebra over $X$ as the set of all admissible pieces of cake, this allows to continue using box-based valuation functions. Thereby, the latter was followed by our argumentation that every box-based valuation function can be uniquely ex-

tended to a measure on $\mathcal{B}(X)$ with the help of Carathéodory's extension theorem. Furthermore, by the means of measure theory we also argued why any larger subset of $\mathfrak{P}(X)$ than $\mathcal{B}(X)$ as a choice for $\mathcal{P}$ could introduce unintended and questionable side effects to cake-cutting.

## 5.2 Publication

P. Kern, D. Neugebauer, J. Rothe, R. Schilling, D. Stoyan, and R. Weishaupt. "Cutting a Cake Is Not Always a "Piece of Cake": A Closer Look at the Foundations of Cake-Cutting Through the Lens of Measure Theory". In: *Social Choice and Welfare* (Submitted)

Preliminary versions of this work were submitted to and accepted at the *8th International Workshop on Computational Social Choice* in 2021 as well as on *arXiv*:

P. Kern, D. Neugebauer, J. Rothe, R. Schilling, D. Stoyan, and R. Weishaupt. "A Closer Look at the Cake-Cutting Foundations through the Lens of Measure Theory". In: *The 8th International Workshop on Computational Social Choice (COMSOC-21)*. Ed. by B. Zwicker and R. Meir. Available online at `https://comsoc2021.net.technion.ac.il/accepted-papers/`. Haifa, Israel: Technion-Israel Institute of Technology, 2021

P. Kern, D. Neugebauer, J. Rothe, R. Schilling, D. Stoyan, and R. Weishaupt. *Cutting a Cake Is Not Always a "Piece of Cake": A Closer Look at the Foundations of Cake-Cutting Through the Lens of Measure Theory.* Tech. rep. arXiv: 2111.05402v1 [cs.GT]. Nov. 2021

## 5.3 Personal Contribution

The writing of this work was conducted jointly with my co-authors Peter Kern, Daniel Neugebauer, Jörg Rothe, René Schilling, and Dietrich Stoyan. Most of the results formulated within this work are a result of close collaboration between all co-authors. However, the initial, technical contributions for Example 1.2, the identification of the set of all admissible pieces of cake with an algebra at the beginning of section 2.1, the formulation of the requirements for valuation functions thereafter, parts (b) and (d) of Theorem 3.1, the formalization of box-based valuation functions, Example 4.3, and Lemma 4.4 were contributed by myself.

# Cutting a Cake Is Not Always a "Piece of Cake": A Closer Look at the Foundations of Cake-Cutting Through the Lens of Measure Theory

Peter Kern[1], Daniel Neugebauer[2], Jörg Rothe[2], René L. Schilling[3], Dietrich Stoyan[4] and Robin Weishaupt[2*]

[1]Mathematical Institute, Heinrich-Heine-Universität Düsseldorf, Düsseldorf, Germany.
[2]Institute for Computer Science, Heinrich-Heine-Universität Düsseldorf, Düsseldorf, Germany.
[3]Institute of Mathematical Stochastics, TU Dresden, Dresden, Germany.
[4]Institute for Stochastics, TU Bergakademie Freiberg, Freiberg, Germany.

*Corresponding author(s). E-mail(s): robin.weishaupt@hhu.de;

**Abstract**

Cake-cutting is a playful name for the fair division of a heterogeneous, divisible good among agents, a well-studied problem at the intersection of mathematics, economics, and artificial intelligence. The cake-cutting literature is rich and edifying. However, different model assumptions are made in its many papers, in particular regarding the set of allowed pieces of cake that are to be distributed among the agents and regarding the agents' valuation functions by which they measure these pieces. We survey the commonly used definitions in the cake-cutting literature, highlight their strengths and weaknesses, and make some recommendations on what definitions could be most reasonably used when looking through the lens of measure theory.

**Keywords:** cake-cutting protocol, admissible piece of cake, finitely additive measure, continuity properties of measures

# 1 Introduction

Since the groundbreaking work of Steinhaus (1948), cake-cutting is a metaphor for the so-called *fair division problem for a divisible, heterogeneous good*, which addresses the problem to split a contested quantity (a "cake") in a fair way among several parties $A, B, C, \ldots$; each party may have its own idea about the value of the different parts of the cake. A traditional way of fair division between two parties $A$ and $B$ would be to let $A$ divide the cake into two pieces (depending on their own valuation) while $B$ has the right to choose one of the pieces, the so-called *cut & choose* protocol. There are other possibilities for two parties as well as extensions to more than two parties (see, e.g., Procaccia, 2016; Lindner and Rothe, 2015, for an overview). Yet, while the basic rules of the game are pretty clear, the assumptions on the actual cutting process are often treated in a gentlemanlike manner. If the whole cake is represented by an interval, say $[0, 1]$, many authors think of the pieces as "intervals," without specifying whether the intervals are open $(a, b) \subset [0, 1]$, half-open $(a, b], [a, b) \subset [0, 1]$, or closed $[a, b] \subseteq [0, 1]$, and how to treat the – possibly twice counted – end points, i.e., $[0, 1/2) \cup [1/2, 1]$ vs. $[0, 1/2] \cup [1/2, 1]$; this is, of course, not an issue if a one-point set like $\{1/2\}$ has zero value for all parties. However, this simple example shows that a formal mathematical approach to cake-cutting needs to address questions like:

- Are (open, closed, half-open) intervals the only possible pieces of cake?
- Do we allow for finitely many or infinitely many cuts (a "cut" being the split of any subset of $[0, 1]$ at a single point)?
- Which properties should a valuation function (by which an agent individually evaluates the pieces of cake) have, and how does it interact with the family of admissible pieces of cake?

For some cases, there is an obvious answer: If we use only finitely many cuts, finite unions of intervals of the form $\langle a, b\rangle$ – where the angular braces indicate either open or closed ends – is all we can get; and if, in addition, any single point $a \in [0, 1]$ has zero value, we do not have to care about the open or closed ends anymore. We will see in Section 2.1 below that this rather implicit assumption brings us in a much more potent framework that can effectively deal with a countably infinite number of cuts.

As soon as we allow for countably infinitely many cuts, things change dramatically, as the following example shows.

*Example 1* (Cantor dust; Cantor's ternary set) Start with the complete cake as a single piece, i.e., $A_0 = [0, 1]$. Now, cut out the middle third of $A_0$ to obtain the intermediate piece $A_1 = A_0 \setminus (1/3, 2/3) = [0, 1/3] \cup [2/3, 1]$ comprising two closed intervals. Next, cut out the middle third of both remaining pieces in $A_1$ to obtain a union of four closed intervals $A_2 = [0, 1/9] \cup [2/9, 1/3] \cup [2/3, 7/9] \cup [8/9, 1]$, see Figure 1. If this procedure is repeated on and on, we will remove countably many open intervals, and the remainder set is $C_{1/3} = \bigcap_{i=1}^{\infty} A_i$. The set $C_{1/3}$ is the **Cantor (ternary) set** (see, e.g., Schilling and Kühn, 2021, § 2.5), and one can show that this is a closed set,

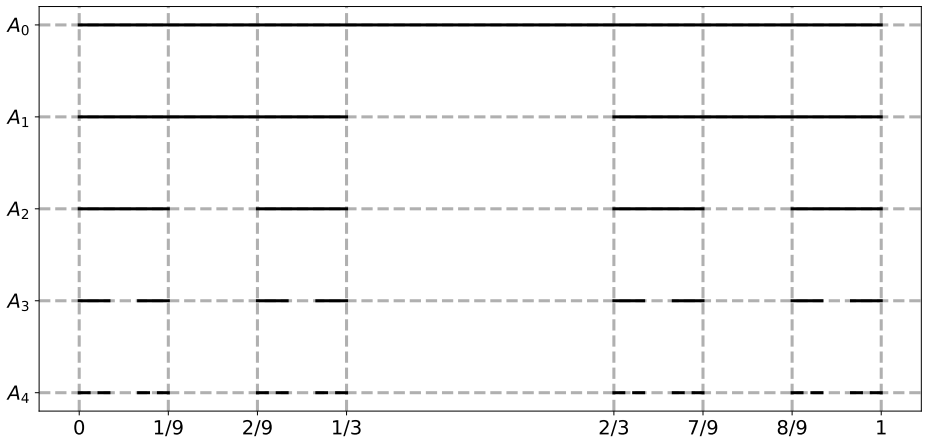



**Fig. 1** Step-wise pieces to be cut for a Cantor-like piece of cake.

which has more than countably many points, does not contain any interval, and is dense in itself, i.e., each of its points is a limit point of a sequence inside $C_{1/3}$. In the usual measuring scale, the original cake had length 1, and the recursively removed pieces have total length

$$\frac{1}{3} + \left(\frac{1}{9} + \frac{1}{9}\right) + \left(\frac{1}{27} + \frac{1}{27} + \frac{1}{27} + \frac{1}{27}\right) + \cdots = \sum_{i \in \mathbb{N}} \frac{2^{i-1}}{3^i} = 1,$$

so that $C_{1/3}$ has zero "length," but it still contains more than countably many points.

The same construction principle, removing at each stage $2^{i-1}$ identical open middle intervals, each having length $p^i$ for some $p$, $0 < p \leq 1/3$, leads to the Cantor set $C_p$, which is, again, closed, uncountable, and does not contain any interval. If, say, $p = 1/4$, the removed intervals have total length $1/2$ and the remaining Cantor dust has "length" $1 - 1/2 = 1/2$. This is not quite expected.

While it is intuitive that the removed intervals should have a certain length, it feels unnatural to speak of the "length" of a dust-like set as $C_p$. In fact, we are dealing here with (one-dimensional) Lebesgue measure, which is the mathematically formal extension of the familiar notion of "length."

An alternative, slightly more formal way of illustrating the Cantor dust is given in the appendix as Example 18.

This example shows that, as soon as we allow for countably many cuts, there can appear sets which may not be written as a countable union of intervals; moreover, although these sets consist of limit points only, they may have strictly positive length.

An important feature of this example is the fact that we extend the family of intervals to a family of subsets which contains (i) finite unions, (ii) countable intersections, and (iii) complements of its members, leading to fairly complicated subsets as, e.g., $C_p$. Moreover, when calculating the length of all removed intervals, we tacitly assumed

- the (finite) additivity of length: The length of two disjoint sets is the sum of their lengths;

- the countable or $\sigma$-additivity which plays the role of a continuity property: The length of a countable union is the limit of the length of the union of the first $N$ sets as $N \to \infty$.

As it will turn out, these are two far-reaching assumptions on the interplay of the valuation function (here: length) with its domain; we will see how this relates to the desirable property that we can cut off pieces of arbitrary length $\ell$, $0 \leq \ell \leq 1$, from the cake $[0, 1]$ (allowing for any valuation values of the cut-off pieces).

Commonly, in cake-cutting theory (see, e.g., Brams and Taylor, 1996; Procaccia, 2016; Lindner and Rothe, 2015) a (piece-wise constant) valuation function $v\colon \mathcal{P} \to [0, 1]$, where $\mathcal{P}$ is some family of subsets of the cake, is represented as shown in Figure 2: The cake $[0, 1]$ is split horizontally into multiple pieces and the number of vertically stacked boxes per piece describes the piece's valuation from some agent's perspective. For example, the valuation function $v$ in Figure 2 evaluates the piece $X' = [0, 2/6]$ with $v(X') = 3/17$. Having this

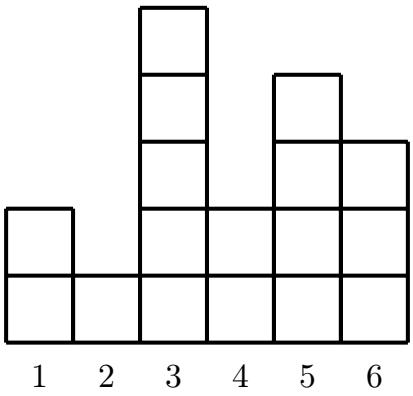



**Fig. 2** Common representation for a valuation function in cake-cutting.

example in mind, one is tempted to assume that $\mathcal{P}$ can always be taken as the power set $\mathcal{P} = \mathfrak{P}([0, 1]) = \{A \mid A \subseteq [0, 1]\}$.

The following classical example from measure theory shows that there cannot exist a valuation function that assigns to intervals $\langle a, b\rangle \subseteq [0, 1]$ their natural length $b - a$, and which is additive, $\sigma$-additive (in the sense explained above), and able to assign a value to every set $A \subseteq [0, 1]$. Things are different if we do not require $\sigma$-additivity (see the discussion in Schilling and Kühn, 2021, § 7.31).

*Example 2* (Vitali, 1905; see also, e.g., Schilling and Kühn, 2021) Let $[0, 1]$ be the standard cake, and assume that the valuation function $v$ is $\sigma$-additive (see Definition 2 on page 8), assigning to any interval its natural length. This means, in particular, that $v$ is invariant under translations and evaluates the complete cake with $v([0, 1]) = 1$. Let us define the relation $*$ as follows: We say that two real numbers $x, y \in \mathbb{R}$ satisfy the relation $*$ if, and only if, $x - y \in \mathbb{Q}$, i.e., their difference is rational. The relation $*$ is an equivalence relation and the corresponding equivalence classes $[x] = \{y \in \mathbb{R} \mid x * y\} \subseteq \mathbb{R}$ lead to a disjoint partitioning of $\mathbb{R}$. By the axiom of

choice, there is a set $V \subset [0,1]$ which contains exactly one representative of every equivalence class $[x]$. A set like $V$ is called a **Vitali set**. Clearly, $V \in \mathfrak{P}([0,1])$ and the sets $q+V = \{q+x \mod 1 \mid x \in V\}$, $q \in \mathbb{Q}$, are a disjoint partition of $[0,1]$; thus $\bigcup_{q\in\mathbb{Q}}(q+V) = [0,1]$. By assumption, $v$ is $\sigma$-additive and assigns to each $q+V$ the same value (translation invariance). Hence, we end up with the contradiction

$$1 = v([0,1]) = v\left(\bigcup_{q\in\mathbb{Q}}(q+V)\right) = \sum_{q\in\mathbb{Q}} v(q+V) = \begin{cases} 0 & \text{if } v(V)=0, \\ \infty & \text{if } v(V)>0. \end{cases}$$

Thus $v$ cannot have the power set of the cake $[0,1]$ as its domain *if we assume that $v$ is $\sigma$-additive*. We will see below that certain commonly used divisibility assumptions are equivalent to the $\sigma$-additivity of the valuation.

*Example 3* (Cantor function) Let us return to Example 1 and interpret the points in the set $C_{1/3}$ as valuable assets which need to be priced. We may assume that the total value of the cake $C_{1/3}$ is 1. We want to construct a "cumulative valuation function $V$" which has the property that for $0 \le a \le b \le 1$ the difference $V(b) - V(a)$ is the value of the points contained in $C_{1/3} \cap (a,b]$. Clearly, $x \mapsto V(x)$ is a (not necessarily strictly) increasing function with $V(0) = 0$ and $V(1) = 1$.

If we agree that the assets should be "homogeneously" priced, then we are automatically led to the following scheme: As the total value of $C_{1/3}$ is one, the value of $C_{1/3} \cap [0, 1/2]$ and $C_{1/3} \cap [1/2, 1]$ should be the same, i.e., $1/2$. Since $C_{1/3} \cap (1/3, 2/3) = \emptyset$, we see that both the left third and the right third of $C_{1/3}$ has the value $1/2$. This means that $V(x) = 1/2$ on the whole middle third $(1/3, 2/3)$.

Now we can repeat this argument in the two remaining sets $C_{1/3} \cap [0, 1/3]$ and $C_{1/3} \cap [2/3, 1]$. Since these pieces are scaled-down versions of the original set $C_{1/3}$, we can repeat our argument to the three thirds of the scaled sets and so we see that the cumulative valuation function $V(x)$ takes the values $1/4$ and $3/4$ on the intervals $(1/9, 2/9)$ and $(7/9, 8/9)$, respectively.

Iterating this procedure *ad infinitum*, the remaining values of $V$ at interfaces of the intervals are uniquely determined by monotonicity and we end up with the so-called *Cantor function* or *devil's staircase*, which is monotone, increasing, continuous, and it is flat (i.e., constant) on all middle-thirds removed in the construction process of $C_{1/3}$ in Example 1; a precise mathematical description can be achieved, e.g., using the alternative representation of the Cantor set in Example 18 of Appendix A, but at this point the pictures in Figure 3 tell it all: The Cantor function is the typical function where the fundamental theorem of integral and differential calculus fails: $V(x)$ is constant on all intervals contained in $[0,1] \setminus C_{1/3}$. Since $V$ is increasing, we can set $V'(x) := \limsup_{h\to 0} \frac{1}{h}(V(x+h) - V(x))$, and this is the usual derivative whenever it exists. In particular, $V'(x) = 0$ on $[0,1] \setminus C_{1/3}$. On the other hand, we have

$$\int_0^x V'(t)\,dt = \int_{C_{1/3}\cap[0,x)} V'(t)\,dt = 0 \neq V(x) - V(0) \quad \text{for any } x \in (0,1].$$

This happens because the "length" of $C_{1/3}$ is zero, i.e., we integrate over "too small a set" (no matter how big the integrand may be, it could even take the value $+\infty$!) so as to pick up any strictly positive value.

The above examples highlight some of the problems when evaluating sets. A Cantor-like piece can only be evaluated if the valuation function is not too

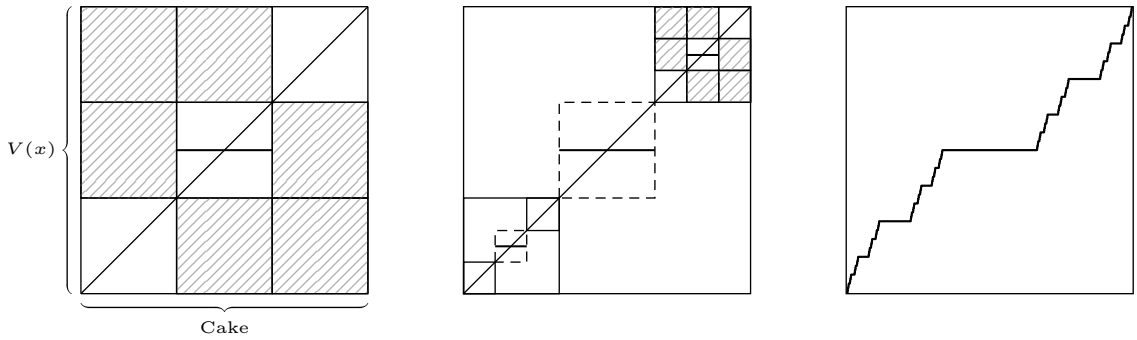



**Fig. 3** Step-by-step construction of the Cantor function: In each step, we subdivide the top right and bottom left squares into nine smaller squares. We keep only the three squares along the diagonal, and discard (gray out) the off-diagonal squares. The middle square is halved by a horizontal line (here $V$ is constant). Repeat.

simplistic. On the other hand, a Vitali set cannot be evaluated at all if we request too many properties of a valuation function, i.e., the domain $\mathfrak{P}([0,1])$ consisting of all possible pieces of cake is, in general, too large.

Across the research field of cake-cutting (see, e.g., the textbooks by Brams and Taylor, 1996; Robertson and Webb, 1998, and the book chapters by Procaccia, 2016; Lindner and Rothe, 2015), there exist several different assumptions on the underlying model. Our goal is to review thoroughly and comprehensively all the different models that are currently applied in the literature. Furthermore, we study the relationships between these models and formulate some related results. It turns out that some of these models are problematic and should not be used as they are formulated. We highlight these models' problems and provide specific examples showing why they are problematic. Our overall goal is to determine a model, which is as simple as possible, yet powerful enough to cope with these problems and still compatible with many of the currently used models.

Frequently, authors proposing cake-cutting protocols abstain from making formal assumptions or from formalizing their model in detail. For example, Brams et al. (1997, p. 553) write:

> *"Many feel that the informality adds to the subject's simplicity and charm, and we would concur. But charm and simplicity are not the only factors determining the direction in which mathematics moves or should move. Our analysis in this paper raises several issues that may only admit a resolution via some negative results. While such results may not require complete formalization of what is permissible, they do appear to require partial versions. We will refer to such partial limitations as theses."*

It would thus be desirable to have some common consensus on which models are useful for any given purpose, and which are not. If we allow only a fixed number of cuts, splitting the cake $[0,1]$ into a finite number of pieces of the type $\langle a,b\rangle \subseteq [0,1]$, a naive approach is always possible: The valuation should be additive and its domain contains unions of finitely many intervals. If, on the other hand, there are potentially infinitely many cuts – e.g., if the players play a game resulting in an *a priori* not fixed number of rounds (such as the finite

unbounded envy-free cake-cutting protocol of Brams and Taylor (1995a)) – the limiting case cannot any longer be treated by a finitely additive valuation and a domain containing only finite unions, see Example 1.

We propose to use ideas from measure theory, which provides the right toolbox to tackle the issues described above. We will see that, at least for the cake $[0,1]$, even the naive approach plus the requirement that we can split every piece $\langle a,b\rangle$ by a single cut into any proportion (in fact, a slightly weaker requirement will do, cf. Definition 2 (D)), automatically leads to the measure-theoretic point of view. That is to say that in many natural situations the naive standpoint is "practically safe" since its obvious shortcomings are automatically "fixed by (measure) theory," if one uses the correct formulation.

# 2 The Rules of the Game

Throughout this paper, $[0,1]$ denotes a standard cake, and the power set $\mathfrak{P}([0,1]) = \{S \mid S \subseteq [0,1]\}$ are all **possible** pieces of cake from a set-theoretic point of view. We define $\mathcal{P} \subseteq \mathfrak{P}([0,1])$ as the set of all **admissible** pieces of $[0,1]$, i.e., those pieces which (a) can be allocated to some players via a cake-cutting protocol, and (b) can be evaluated by the players using their valuation functions. Sometimes it is necessary to consider an "abstract" cake $X$, with its possible and admissible pieces $\mathfrak{P}(X)$ and $\mathcal{P} \subseteq \mathfrak{P}(X)$. Some results for the standard cake $[0,1]$ remain true for abstract cakes. For example, an abstract cake $X$ might be contained in the $n$-dimensional unit cube: $X \subseteq [0,1]^n$.

## 2.1 Dividing a Cake with Finitely Many Cuts

We start by formulating requirements for $\mathcal{P}$ regarding the admissible pieces of cake. The discussion in this section applies both to the standard cake $[0,1]$ and the abstract cake $X$. Obviously, we want to be able to allocate the complete cake $X$ as well as an empty piece $\emptyset$ to a player and therefore, $X \in \mathcal{P}$ and $\emptyset \in \mathcal{P}$ must hold. If $A \subseteq X$ is already allocated to some player, i.e., $A \in \mathcal{P}$, then we want to be able to give the remainder of the cake to another player; so for all $A \in \mathcal{P}$, we demand that the complement of $A$, denoted by $\overline{A} = X \setminus A$, is in $\mathcal{P}$. Furthermore, we want to be able to cut and combine pieces of cake; so for all $A, B \in \mathcal{P}$, we require $A \cup B \in \mathcal{P}$. Note that $A \cap B = \overline{\overline{A} \cup \overline{B}}$ and $A \setminus B = A \cap \overline{B}$, so our previously formulated requirements also allow us to allocate the intersection of a finite number of pieces of cake and to evaluate the difference of two pieces of cake.

**Definition 1** Let $X$ be a(n abstract) cake. A family $\mathcal{A} \subseteq \mathfrak{P}(X)$ is called an **algebra** over $X$ if $\emptyset \in \mathcal{A}$ and for all $A, B \in \mathcal{A}$ it holds that $\overline{A}$ and $A \cup B \in \mathcal{A}$.

It is worth noting that only by the formulation of intuitive requirements with respect to the set of all admissible pieces of cake, we ended up with a well-studied, structured concept from measure theory: an algebra.

*Example 4* If $X = [0,1]$, then $\mathfrak{P}(X)$ and $\{\emptyset, X\}$ are algebras – in fact these are the largest possible and the smallest possible algebras over $[0,1]$. Another useful algebra is the family $\mathcal{I}([0,1])$ of all unions of finitely many intervals in $[0,1]$ – and it is easy to check that $\mathcal{I}([0,1])$ is the smallest algebra containing all closed (or all open or all half-open) intervals from $[0,1]$. While it is obvious that $\{\emptyset, [0,1]\}$ is useless for our purpose, as then only two possible pieces can be allocated, the complete cake and an empty piece, we might – at the other extreme – also take $\mathfrak{P}([0,1])$ as the set for the admissible pieces of $[0,1]$. However, when choosing $\mathcal{P}$, we must also ensure that meaningful valuation functions can exist for this set, and Example 2 shows that for a rather natural valuation function – geometric length – $\mathfrak{P}([0,1])$ is too big.

Let us list the common requirements for the players' valuation functions. A **valuation function** $v$ shall assign to any admissible piece of cake $A \in \mathcal{P}$ some nonnegative real number, i.e., $v\colon \mathcal{P} \to [0,\infty]$. In order to normalize the players' valuations and keep them comparable, we demand that $v(\emptyset) = 0$ and $v(X) = 1$ hold. Hence, we can further limit the valuation function's range to $[0,1]$, i.e., we have $v\colon \mathcal{P} \to [0,1]$. The next definition lists further requirements for a valuation function.

**Definition 2** Let $X$ be a(n abstract) cake and $\mathcal{A}$ the algebra of admissible pieces. A **valuation function** is a function $v\colon \mathcal{A} \to [0,1]$, which is normalized, i.e., $v(\emptyset) = 0$ and $v(X) = 1$. Moreover, $v$ is called

(M) **monotone** if for $A, B \in \mathcal{A}$ with $A \subseteq B$, one has $v(A) \leq v(B)$;
(A) **additive** or **finitely additive** if for all $A, B \in \mathcal{A}$ such that $A \cap B = \emptyset$, one has $v(A \cup B) = v(A) + v(B)$;
(Σ) **$\sigma$-additive** or **countably additive** if for any sequence $(A_n)_{n\in\mathbb{N}}$ of pieces in $\mathcal{A}$ such that $A_i \cap A_j = \emptyset$ $(i \neq j)$ and $\bigcup_{i\in\mathbb{N}} A_i \in \mathcal{A}$, one has $v(\bigcup_{i\in\mathbb{N}} A_i) = \sum_{i\in\mathbb{N}} v(A_i)$;
(D) **divisible** if for every $A \in \mathcal{A}$ and for every real number $\alpha$, $0 \leq \alpha \leq 1$, there exists some $A_\alpha \in \mathcal{A}$ with $A_\alpha \subseteq A$ such that $v(A_\alpha) = \alpha v(A)$.

Clearly, (Σ) implies (A) – take $A_1 = A$, $A_2 = B$, and $A_i = \emptyset$ for $i \geq 3$ – and (A) is equivalent to the so-called **strong additivity**, defined as $v(A \cup B) = v(A) + v(B) - v(A \cap B)$: Just observe that $A \cup B = [A \setminus (A \cap B)] \cup [B \setminus (A \cap B)] \cup [A \cap B]$, i.e., $A \cap B \neq \emptyset$ counts towards both $v(A)$ and $v(B)$ but only once in $v(A \cup B)$, hence the correction $-v(A \cap B)$. Finally, (strong) additivity implies monotonicity.

The assumption that $\mathcal{A}$ is an algebra makes sure that we can indeed perform all of the above manipulations with sets without ever leaving $\mathcal{A}$. Note, however, that (Σ) and (D) impose further assumptions on the structure of $\mathcal{A}$.

*Remark 5* Let $X$ be a(n abstract) cake and $\mathcal{A} \subseteq \mathfrak{P}(X)$ an algebra over $X$. Any additive valuation is a **finitely additive measure** with total mass $v(X) = 1$ (see, e.g., Schilling, 2017, Chapter 4).

Requirement (D) not only demands more from $\mathcal{A}$ but also from $v$. Specifically, (D) entails that any $N \in \mathcal{A}$ which does not contain a nonempty and strictly smaller piece of cake – this is an **atom**, i.e., an indivisible piece of cake – must have zero valuation.

**Definition 3** Let $\mathcal{A}$ be an algebra over a(n abstract) cake $X$ and $v$ be a finitely additive valuation. A set $A \in \mathcal{A}$ is an **atom** if $v(A) > 0$ and every $B \subseteq A$, $B \in \mathcal{A}$, satisfies $v(B) = \alpha v(A)$ with $\alpha = 0$ or $\alpha = 1$.

Clearly, a valuation $v$ which enjoys property (D) cannot have atoms.

## 2.2 Dividing the Standard Cake

Let us briefly discuss the consequences of the notions introduced in the previous section if $X$ is the standard cake $[0, 1]$. If, in addition, $\mathcal{A}$ contains all intervals of type $\langle a, b\rangle$, then all singletons $\{a\} = [a, b] \setminus (a, b]$ are in $\mathcal{A}$, and they are the only possible atoms. In this case, (D) entails that $v$ does not charge single points: $v(\{a\}) = 0$ for all $a \in [0, 1]$. This is the proof of the following lemma.

*Lemma 6* Let $[0, 1]$ be the standard cake and $\mathcal{A}$ an algebra of admissible sets. Every additive valuation function $v\colon \mathcal{A} \to [0, 1]$ that satisfies (D) is atom-free. In particular, if $\mathcal{A} \supset \mathcal{I}([0, 1])$ contains all intervals, then $v(\{a\}) = 0$ for all $a \in [0, 1]$.

Quite often, we require valuation functions to satisfy **continuity**, a property that is crucial for so-called *moving-knife* cake-cutting protocols to work.

**Definition 4** Let $v\colon \mathcal{A} \to [0, 1]$ be a finitely additive valuation function on the algebra $\mathcal{A} = \mathcal{I}([0, 1])$ of finite unions of intervals from $[0, 1]$.

1. The function $x \mapsto F_v(x) := v([0, x])$, $x \in [0, 1]$, is the **distribution function** of the valuation $v$.
2. The valuation $v$ is said to be **continuous** if $x \mapsto F_v(x)$ is continuous.

Since $v$ is additive, $F_v\colon [0, 1] \to [0, 1]$ is positive, monotonically increasing, and bounded by $F_v(1) = 1$. Note that a continuous valuation function on $\mathcal{I}([0, 1])$ cannot have atoms, as

$$v(\{x\}) = v([0, x] \setminus [0, x)) = F(x) - F(x-) = 0, \quad \text{where} \quad F(x-) = \lim_{y \uparrow x} F(y).$$

The continuity of $v$ can also be cast in the following way: For all $a$ and $b$ with $0 \leq a < b \leq 1$ satisfying $v([0, a]) = \alpha$ and $v([0, b]) = \beta$, and for every $\gamma \in [\alpha, \beta]$, there exists some $c \in [a, b]$ such that $v([0, c]) = \gamma$. This explains the close connection between continuity and divisibility of $v$. In fact, assuming

divisibility (D) of $v$, it can be shown that the distribution function is necessarily continuous. The following proof of this statement is inspired by Schilling and Stoyan (2016, Example 3.4).

*Lemma 7* Let $v$ be an additive valuation for the standard cake $[0,1]$, where $\mathcal{I}([0,1])$ denotes the family of admissible pieces. If $v$ is divisible, then the distribution function $F = F_v$ is a continuous function with $F(0) = 0$.

*Proof* We have seen in Lemma 6 that a divisible additive valuation $v$ has no atoms, so $F(0) = v(\{0\}) = 0$. Since $F$ is monotone and bounded, the one-sided limits $F(t-) := \lim_{s\uparrow t} F(s)$ and $F(u+) := \lim_{s\downarrow u} F(s)$ exist for all $t \in (0,1]$ and $u \in [0,1)$.

Assume that $F$ is not continuous. Then there exists some $t_0 \in [0,1]$ such that $F(t_0-) < F(t_0)$ or $F(t_0+) > F(t_0)$. If $F(t_0) - F(t_0-) = \varepsilon > 0$, then there exists some $t_1 < t_0$ such that $F(t_0) - F(t_1) \leq \frac{3}{2}\varepsilon$. Set $I := (t_1, t_0]$ and observe that $v(I) = F(t_0) - F(t_1) \in \left[\varepsilon, \frac{3}{2}\varepsilon\right]$. Pick an arbitrary $J \in \mathcal{I}([0,1])$ which is contained in $I$. Since $J$ is a finite union of intervals, $J$ differs from its closure $\bar{J}$ by at most finitely many points; as $v(\{x\}) = 0$ for any $x \in [0,1]$, we have $v(J) = v(\bar{J})$.

We distinguish two cases: If $t_0 \in \bar{J}$ is not an isolated point, then $v(J) = v(\bar{J}) \geq \varepsilon$. If $t_0 \notin \bar{J}$ or if $t_0 \in \bar{J}$ is an isolated point, then we have due to $v(\{t_0\}) = 0$ that

$$\begin{aligned} v(J) = v(\bar{J}) &\leq F(t_0-) - F(t_1) = \big(F(t_0) - F(t_1)\big) - \big(F(t_0) - F(t_0-)\big) \\ &= v(I) - \varepsilon \leq \tfrac{1}{2}\varepsilon. \end{aligned}$$

Hence, it is not possible to select a piece of cake $J \in \mathcal{I}([0,1])$ with $J \subseteq I$ and $v(J) = \frac{3}{4}\varepsilon \in \left[\frac{1}{2}\cdot v(I), \frac{3}{4}\cdot v(I)\right]$, which contradicts divisibility.

If $F(t_0+) - F(t_0) = \varepsilon > 0$, a similar argument applies. □

Conversely, if the distribution function $F_v$ of a finitely additive valuation $v$ defined on $\mathcal{I}(X)$ is continuous with $F_v(0) = 0$, then it is easy to see that $v$ is divisible. Hence we get:

*Corollary 8* A finitely additive valuation $v$ on $\mathcal{I}([0,1])$ is divisible if, and only if, its distribution function $F_v$ is continuous with $F_v(0) = 0$. This is also equivalent to $v$ being atom-free.

Corollary 8 establishes a one-to-one correspondence between divisible valuations and monotonically increasing, continuous functions on $[0,1]$ which are 0 at the origin and 1 at $x = 1$. This shows that $x \mapsto x$ is a valuation (it assigns every interval $\langle a, b\rangle$ its natural length $b - a$) but also the Cantor function $V(x)$ from Example 3 can be viewed as a valuation function.

We will see in the next section that every finitely additive, divisible valuation can be extended to become and identified with a unique $\sigma$-additive measure that is defined on the Borel $\sigma$-algebra $\mathcal{B}(X)$; this is the smallest family of sets that contains all intervals and that is stable under complements and countable unions of its members. This enables us to evaluate sets in $\mathcal{B}(X)$ that are not finite unions of intervals, such as the Cantor set in Example 1.

## 2.3 Measure Theory: The Art of Dividing a Cake by Countably Many Cuts

Up to now we have only allowed finitely many cuts when dividing the cake. But we may easily come into the situation where the number of cuts is not limited; not all protocols in the cake-cutting literature are finite. Thus we are led to consider unions of countably many pieces and the valuation of such countable unions, see also property ($\Sigma$) in Definition 2. To deal with such situations, measure theory provides the right tools.

We will now introduce some basics from measure theory, which we need in the subsequent discussion of the cake-cutting literature. Our standard references for measure theory are the monographs by Schilling (2017) and Schilling and Kühn (2021), where also further background information can be found.

**Definition 5** Let $X$ be a(n abstract) cake. A subset $\mathcal{A} \subseteq \mathfrak{P}(X)$ is called a ***σ*-algebra** over $X$ if $\mathcal{A}$ is an algebra over $X$ and, for all sequences $(A_n)_{n\in\mathbb{N}}$ with $A_n \in \mathcal{A}$, the countable union $\bigcup_{n\in\mathbb{N}} A_n$ is in $\mathcal{A}$, too.

Let us return to the standard cake $[0,1]$. Every algebra in $[0,1]$ containing finitely many sets is automatically a $\sigma$-algebra. On the other hand, $\mathfrak{P}([0,1])$ is both an algebra and a $\sigma$-algebra, whereas the family $\mathcal{I}([0,1])$ is an algebra, but not a $\sigma$-algebra: For instance, the Cantor dust $C_p$ (cf. Example 1) is not in $\mathcal{I}([0,1])$. Recall that we defined $\mathcal{I}([0,1])$ to be the smallest algebra containing all (finite unions of) intervals in $[0,1]$; thus it is natural to consider the smallest $\sigma$-algebra containing all (finite unions of) intervals in $[0,1]$.

To see that this is well-defined, we need a bit more notation. Recall that $\langle a,b\rangle$ stands for any (open, closed, or half-open) interval of $[0,1]$. We denote by

$$\mathcal{Q}([0,1]) = \{\langle a,b\rangle \mid a,b \in [0,1]\}\,.$$

the *family of all intervals within* $[0,1]$.

Moreover, if $\mathcal{P} \subseteq \mathfrak{P}([0,1])$ is any family, then $\sigma(\mathcal{P})$ denotes the smallest $\sigma$-algebra containing $\mathcal{P}$. This can be a fairly complicated object and its existence is not really obvious. To get an idea as to why $\sigma(\mathcal{P})$ makes sense, we note that $\mathcal{P} \subseteq \mathfrak{P}([0,1])$, that $\mathfrak{P}([0,1])$ is a $\sigma$-algebra, and that the intersection of any number of $\sigma$-algebras is still a $\sigma$-algebra.

The next lemma is a standard result from measure theory.

*Lemma 9* Let $\mathcal{P}$ denote any of the four families of open intervals, closed intervals, left-open intervals, or right-open intervals within $[0,1]$. It holds that

$$\sigma(\mathcal{P}) = \sigma(\mathcal{Q}([0,1])).$$

The fact that $\sigma(\mathcal{Q}([0,1]))$ coincides with the $\sigma$-algebra generated by all closed intervals in $[0,1]$ can be used to generalize Lemma 9 to abstract cakes,

which carry a topology, hence a family of open and of closed sets. The thus generated "topological" $\sigma$-algebra plays a special role and has a special name.

**Definition 6** We denote by $\mathcal{B}([0,1])$ the smallest $\sigma$-algebra on $[0,1]$ containing all closed intervals from $[0,1]$ and call it the **Borel** or **topological $\sigma$-algebra** over $[0,1]$.

The following definition is also well-known. We state it only for the standard cake, but it is clear how to extend it to abstract cakes.

**Definition 7** Let $[0,1]$ be a cake and $\mathcal{A}$ a $\sigma$-algebra on $[0,1]$. A (positive) **measure** $\mu$ on $[0,1]$ is a map $\mu\colon \mathcal{A} \to [0,\infty]$ satisfying that $\mu(\emptyset) = 0$ and $\mu$ is $\sigma$-additive.

It is useful to see a measure $\mu$ as a function defined on the sets. If the set-function is additive, then $\sigma$-additivity is, in fact, a continuity requirement on $\mu$, as it allows to interchange the limiting process in the infinite union $\bigcup_{i\in\mathbb{N}} A_i$ of pairwise disjoint sets with a limiting process in the sum. To wit:

$$\mu\left(\bigcup_{n\in\mathbb{N}} A_n\right) = \lim_{N\to\infty} \mu\left(\bigcup_{n=1}^{N} A_n\right) = \lim_{N\to\infty} \sum_{n=1}^{N} \mu\left(A_n\right) = \sum_{n\in\mathbb{N}} \mu\left(A_n\right);$$

since all terms are positive, there is no convergence issue. Equivalently, we can state $\sigma$-additivity as $B_1 \subset B_2 \subset B_3 \subset \cdots \uparrow B = \bigcup_{n\in\mathbb{N}} B_n$, then $\mu(B_n) \uparrow \mu(B)$ (for any measure $\mu$) or as $C_1 \supset C_2 \supset C_3 \supset \cdots \downarrow C = \bigcap_{n\in\mathbb{N}} C_n$, then $\mu(C_n) \downarrow \mu(C)$ (for finite measures $\mu$).

Sometimes (and a bit provocatively) it is claimed that there are essentially only two measures on $[0,1]$ (or on $\mathbb{R}$ or $\mathbb{R}^n$): **Lebesgue measure** $A \mapsto \lambda(A)$ and **Dirac measure** $A \mapsto \delta_x(A)$, where $x \in [0,1]$ is a fixed point. Let us briefly discuss these two extremes and explain as to why the claim is incorrect but still sensible.

***Dirac Measure.***

Let $a \in [0,1]$ be a fixed point and set $A \mapsto \delta_a(A) = 1$ or $= 0$ according to $a \in A$ or $a \notin A$, respectively. This definition works for any $A \subseteq [0,1]$, and it is easy to see that this set-function is indeed a measure (in the sense of Definition 7 on the $\sigma$-algebra $\mathcal{A} = \mathfrak{P}([0,1])$ – or any smaller $\sigma$-algebra over $[0,1]$.

Dirac's measure is the derivative of the physicists' "Delta function": Indeed, the integral $\int f(x)\,\delta_a(dx)$ can be shown to yield $f(a)$, which makes $\delta_a(x) := \frac{d}{dx}\delta_a$ (the derivative is understood in a distributional sense) a "function" such that "$\delta_a(x) = 0$ if $x \neq a$, $\delta_a(a) = \infty$, and $\int \delta_a(x) = 1$" – the trouble being that $\delta_a(x)$ cannot be defined pointwise for each $x \in X$. This is best understood if we look at the distribution function: This is the Heaviside function $\delta_a([0,x]) = \mathbf{1}_{[a,1]}(x)$ on $X$, which is zero on $[0,a)$, one on $[a,1]$ with a jump of size 1, and exploding differential quotient at $x = a$.

We call $\{a\}$ the **support** of $\delta_a$ since, by definition, $\delta_a$ charges only sets such that $\{a\} \subseteq A$. If we compare Dirac measure with Lebesgue's measure, the problem is that the support of $\delta_a$ is a degenerate interval $\{a\} = [a, a]$ of length zero, see below.

***Lebesgue Measure.***

The idea behind Lebesgue measure is to have a set-function $A \mapsto \lambda(A)$ in $[0, 1]$ (or in $\mathbb{R}$ or $\mathbb{R}^n$) with all properties of the familiar volume from geometry; in particular, we want a volume that is additive and invariant under shifts and rotations. Thus it is natural to define for a simple set $Q$ like an interval $Q = (a, b] \subset [0, 1]$ (or an $n$-dimensional "cube" $Q = \overset{n}{\underset{i=1}{\times}} (a_i, b_i]$)

$$\lambda(Q) = b - a$$

$$\left( \text{respectively,} \quad \lambda(Q) = \prod_{i=1}^{n} (b_i - a_i) = \text{length} \times \text{width} \times \text{height} \times \cdots \right).$$

Invariance under shifts together with the $\sigma$-additivity ($\Sigma$) allow us to exhaust ("triangulate") more complicated shapes like a circle with countably many disjoint sets $(Q_n)_{n\in\mathbb{N}}$ such that with $A = \bigcup_{n\in\mathbb{N}} Q_n$, we have $\lambda(A) = \sum_{n\in\mathbb{N}} \lambda(Q_n)$. The restriction to countable unions is natural, as we exhaust a given shape by nontrivial sets $Q_n$, having nonempty interior: Each of them contains a rational point $q \in \mathbb{Q}^n$; hence, there are at most countably many nonoverlapping $Q_n$.

There are immediate questions with this approach: Which types of sets can be "measured"? Is the procedure unique? Is the process of measuring more complicated sets constructive? At this point we encounter a problem: General sets $A \subseteq \mathbb{R}^n$ are way too complicated to get a well-defined and unique extension of $\lambda$ from the rectangles to $\mathfrak{P}(\mathbb{R}^n)$. In dimension $n = 1$ and for the standard cake $[0, 1]$, the Cantor sets $C_p$ from Example 1 were already challenging, but the Vitali set from Example 2 shows that the cocktail of shift invariance and $\sigma$-additivity becomes toxic.

The way out is the notion of measurable sets and Carathéodory's extension theorem (stated as Theorem 10 further down). This works as follows: In view of the $\sigma$-additivity property of $\lambda$, it makes sense to consider the $\sigma$-algebra $\mathcal{A} \subseteq \mathfrak{P}(\mathbb{R}^n)$ which contains the intervals (respectively, cubes). Thus we naturally arrive at the notion of the Borel $\sigma$-algebra as the canonical domain of Lebesgue measure. Unfortunately, there are so many Borel sets that we cannot build them constructively from rectangles – we would need transfinite induction for this – and this is one of the reasons why cutting a cake is not always a piece of cake.

The question of whether *every* set $A \subseteq \mathbb{R}^n$ has a unique geometric volume (in the above sense) is dimension-dependent. If $n = 1$ or $n = 2$, we can extend the notion of length and area to all sets, but not in a unique way. In dimension 3 and higher, we'll end up with contradictory statements (such as the *Banach–Tarski paradox*; see, e.g., Wagon, 1985) if we try to have a finitely additive

geometric volume for all sets. This conundrum can be resolved by looking at the Borel sets or the Lebesgue sets – these are the Borel sets enriched by all subsets of Borel sets with Lebesgue measure zero.

***General Measures.***

Let us return to the assertion that $\lambda$ and $\delta_a$ are "essentially the only measures" on $\mathbb{R}^n$. To keep things simple, we discuss here only the standard cake $[0,1]$.

Lebesgue's decomposition theorem shows that all $\sigma$-additive measures $\mu$ on $[0,1]$ with the Borel $\sigma$-algebra $\mathcal{B}([0,1])$ are of the form $\mu = \mu^{\mathrm{ac}} + \mu^{\mathrm{sc}} + \mu^{\mathrm{d}}$ where "ac," "sc," and "d" stand for absolutely continuous, singular continuous, and discontinuous. This is best explained by looking at the distribution function $F(x) = F_\mu(x)$. Since $x \mapsto F(x)$ is increasing, it is either continuous or discontinuous (with at most countably many discontinuities), accounting for the parts $(\mu^{\mathrm{ac}}, \mu^{\mathrm{sc}})$ and $\mu^{\mathrm{d}}$, respectively. At the points where $F$ is continuous, we have again two possibilities: $F$ is either differentiable ($F'(x) = f(x)$) or it isn't, yielding "ac" vs. "sc." From Lebesgue's differentiation theorem it is known that the points with "sc" or "d" must have Lebesgue measure zero. Thus, we finally arrive at the decomposition

$$\mu(dx) = f(x)\,dx + \mu^{sc}(dx) + \sum_i (F(x_i) - F(x_i-))\,\delta_{x_i}(dx), \tag{1}$$

where $x_1, x_2, \ldots$ are the at most countably many discontinuities (jump points) of $F$ and $f(x) = \frac{d}{dx}F(x)$.

Here are four typical **examples** for valuations corresponding to these cases.

- Purely **ac**: $F^{\mathrm{ac}}(x) := x$ is absolutely continuous since $\frac{d}{dx}F^{\mathrm{ac}}(x) = 1$ exists and $F^{\mathrm{ac}}(x) = \int_0^x 1\,dt$. This $F^{\mathrm{ac}}$ corresponds to Lebesgue measure. In general, an absolutely continuous $F^{\mathrm{ac}}(x)$ is always of the form $F^{\mathrm{ac}}(x) - F^{\mathrm{ac}}(0) = \int_0^x f(t)\,dt$ and $f(t) = \frac{d}{dt}F^{\mathrm{ac}}(t)$.
- Purely **sc**: The Cantor function $F^{\mathrm{sc}}(x) := V(x)$ from Example 3 is continuous, but it is not absolutely continuous. $V'(x)$ exists (in a classical sense) only in the points $[0,1] \setminus C_{1/3}$ and $V(x) \neq \int_0^x V'(t)\,dt$. The corresponding valuation is nevertheless of the form $v((a,b]) = V(b) - V(a)$, but it cannot be represented in the form $\int_a^b f(t)\,dt$ for any function $f$.
- Purely **d**: Any increasing step-function with jumps of size $\Delta_i > 0$, $i = 1, 2, \ldots$, at the points $x_i \in [0,1]$ corresponds to the discontinuous case: We have atoms exactly at the points $x_i$ where $F^{\mathrm{d}}(x)$ is discontinuous (i.e., jumps). The general form of such functions is $F^{\mathrm{d}}(x) = \sum_{i=1}^{\infty} \Delta_i \mathbf{1}_{[x_i,1]}(x)$ where $\sum_{i=1}^{\infty} \Delta_i = 1$.
- Mixed **ac**+**cs**+**d**: Let $p_{\mathrm{ac}}, p_{\mathrm{sc}}, p_{\mathrm{d}} \in [0,1]$ be such that $p_{\mathrm{ac}} + p_{\mathrm{sc}} + p_{\mathrm{d}} = 1$ and let $F^{\mathrm{ac}}, F^{\mathrm{sc}}, F^{\mathrm{d}}$ be as in the previous examples. Then the convex combination $F(x) = p_{\mathrm{ac}}F^{\mathrm{ac}}(x) + p_{\mathrm{sc}}F^{\mathrm{sc}}(x) + p_{\mathrm{d}}F^{\mathrm{d}}(x)$ corresponds to a valuation which combines all three types of (dis-)continuity properties.

Let us close this section with the central result on the extension of valuations defined on an algebra $\mathcal{A}$ to measures on the $\sigma$-algebra $\sigma(\mathcal{A})$ generated by $\mathcal{A}$. We state it only for the standard cake; the formulation for more abstract cakes is obvious.

**Theorem 10** (Carathéodory's extension theorem) *Let $v$ be a valuation on $[0,1]$ and denote by $\mathcal{A}$ the algebra of admissible pieces of cake. If $v$ is additive and $\sigma$-additive relative to $\mathcal{A}$, i.e., $v$ satisfies $(\Sigma)$, then there is a unique extension of $v$, defined on $\sigma(\mathcal{A})$, which is a $\sigma$-additive measure on $\sigma(\mathcal{A})$.*

## 2.4 Abstract Cakes

Let us briefly discuss more general cakes $X$ than $[0,1]$. In this section, $X \neq \emptyset$ will be a general set, $\mathcal{A}$ an algebra of admissible pieces, and $\sigma(\mathcal{A})$ the $\sigma$-algebra generated by $\mathcal{A}$. The definition and the properties of a valuation $v\colon \mathcal{A} \to [0,1]$ (cf. Definition 2) still work in this general setting, but since $X$ is abstract, there may not be (an equivalent of) a distribution function; this means that the connection between divisibility and $\sigma$-additivity, cf. Lemma 7 and Corollary 8, might fail in an abstract setting.

We begin with a new definition of (D) for finitely additive valuations on abstract cakes.

**Definition 8** A finitely additive valuation $v$ on an abstract cake $X$ and an algebra of admissible pieces $\mathcal{A}$ has the property (DD) if for every $A \in \mathcal{A}$ and $\alpha \in (0,1)$, there is an increasing sequence of sets $B_\alpha^1 \subset B_\alpha^2 \subset B_\alpha^3 \subset \cdots$, $B_\alpha^n \in \mathcal{A}$, such that $B_\alpha^n \subset A$ and $\sup_{n\in\mathbb{N}} v(B_\alpha^n) = \alpha v(A)$.

Property (DD) essentially says that for every value $\alpha v(A) \in [0,1]$ we can find an admissible piece of cake $B_\alpha^n \subset A$ whose valuation $v(B_\alpha^n)$ is close to $\alpha v(A)$. The limiting piece $\bigcup_n B_\alpha^n$, which should produce the value $\alpha v(A)$ exactly, may not be admissible if we are restricted to finitely many cuts.

If $v$ is a $\sigma$-additive valuation and $\mathcal{A}$ a $\sigma$-algebra, then $B_\alpha := \bigcup_{n\in\mathbb{N}} B_\alpha^n$ is again in $\mathcal{A}$, and, because of $\sigma$-additivity, we see that $v(B_\alpha) = \sup_{n\in\mathbb{N}} v(B_\alpha^n)$. Thus the properties (D) and (DD) are indeed equivalent for $\sigma$-additive valuations (or, in view of Corollary 8, for finitely additive valuations on the standard cake $[0,1]$ and $\mathcal{A} \supset \mathcal{I}([0,1])$).

We will also need the opposite of the property (DD); to this end, recall Definition 3 of an atom. If $A$ and $B$ are atoms, then we have either $v(A\cap B) = 0$ or $v(A \cap B) = v(A) = v(B) > 0$; in the latter case, if $v(A \cap B) > 0$, we call the atoms *equivalent*. If $A$ and $B$ are nonequivalent, then $A$ and $B \setminus A$ are still nonequivalent and disjoint. Iterating this procedure, we can always assume that countably many nonequivalent atoms $(A_n)_{n\in\mathbb{N}}$ are disjoint: Just replace the atoms by $A_1, A_2 \setminus A_1, \ldots, A_{n+1} \setminus \bigcup_{i=1}^n A_i, \ldots$.

Since $v(X) = 1$, a finitely additive valuation $v$ can have at most $n$ nonequivalent atoms such that $v(A) \geq \frac{1}{n}$, and so there are at most countably many

atoms. Comparing Definition 8 which defines property (DD) with Definition 3 of an atom, it is clear that (DD) implies that $v$ has no atoms. We will see in Theorem 11 that the converse implication holds as well.

**Definition 9** Let $v$ be a finitely additive valuation on the algebra $\mathcal{A}$ over a(n abstract) cake $X$. The valuation $v$ is **sliceable** if for any $\varepsilon > 0$, there are finitely many disjoint sets $B_i \in \mathcal{A}$, $i = 1, \dots, n$, $n = n(\varepsilon)$, such that $0 < v(B_i) \leq \varepsilon$ and $X = B_1 \cup \dots \cup B_n$.

A set $B \in \mathcal{A}$ is ***v*-sliceable** if the set-function $A \mapsto v(A \cap B)$ is sliceable.

We will now see that a sliceable finitely additive valuation enjoys property (DD), and *vice versa*, i.e., sliceability, atom-freeness, and property (DD) are pairwise equivalent for finitely additive valuations.

**Theorem 11** *Let $v$ be a finitely additive valuation on an algebra $\mathcal{A}$ over a(n abstract) cake $X$. The conditions* (DD)*, "$v$ is sliceable," and "$v$ has no atoms" are pairwise equivalent.*

*Proof* We start by showing that atom-freeness implies sliceability. Fix $\varepsilon > 0$.

**Step 1:** Let $Y \subseteq X$ be *any* subset, and assume that there is some $B \subseteq Y$, $B \in \mathcal{A}$, such that $v(B) > 0$. Define

$$\mathcal{F}^Y := \mathcal{F}^Y_\varepsilon := \{F \in \mathcal{A} \mid F \subseteq Y,\ 0 < v(F) \leq \varepsilon\}.$$

We claim that for the special choice $Y = B \in \mathcal{A}$ the family $\mathcal{F}^B$ is not empty.

Since $B$ is not an atom, there is some $F \subseteq B$, $F \in \mathcal{A}$, with $0 < v(F) < v(B)$.

If $v(F) \leq \varepsilon$, then $F \in \mathcal{F}^B$, and we are done.

If $v(F) > \varepsilon$, we assume, to the contrary that there is no subset $F' \subseteq F$, $F' \in \mathcal{A}$, with $0 < v(F') \leq \varepsilon$. Since $F$ cannot be an atom, there is a subset $F' \subseteq F$ with $\varepsilon < v(F') < v(F)$ and $v(F \setminus F') > \varepsilon$. Iterating this with $F \rightsquigarrow F \setminus F'$ furnishes a sequence of disjoint sets $F_1 = F', F_2, F_3, \dots$ with $v(F_i) > \varepsilon$ for all $i \in \mathbb{N}$. This is impossible since $v(F) < \infty$. So we can find some $F' \subseteq F \subseteq B$ with $0 < v(F') \leq \varepsilon$, i.e., $\mathcal{F}^B$ is not empty.

**Step 2:** Define a(n obviously monotone) set-function $c(Y) := \sup_{C \in \mathcal{F}^Y} v(C)$ for any $Y \subseteq X$; as usual, $\sup \emptyset = 0$. Since $\mathcal{F}^X$ is not empty, we can pick some $B_1 \in \mathcal{F}^X$ such that $\frac{1}{2}c(X) < v(B_1) \leq \varepsilon$.

If $v(X \setminus B_1) \leq \varepsilon$, we set $B_2 := X \setminus B_1$; otherwise, we can pick some $B_2 \in \mathcal{F}^{X \setminus B_1}$ such that $\frac{1}{2}c(X \setminus B_1) < v(B_2) \leq \varepsilon$.

In general, if $v(X \setminus (B_1 \cup \dots \cup B_n)) \leq \varepsilon$, we set $B_{n+1} = X \setminus (B_1 \cup \dots \cup B_n)$; otherwise, we pick

$$B_{n+1} \in \mathcal{F}^{X \setminus (B_1 \cup \dots \cup B_n)} \quad \text{such that} \quad \frac{1}{2}c(X \setminus (B_1 \cup \dots \cup B_n)) \leq v(B_{n+1}) \leq \varepsilon. \tag{2}$$

We are done if this procedure stops after finitely many steps; otherwise, we get a sequence of disjoint sets $B_1, B_2, \dots$ satisfying (2). Define $B_\infty := X \setminus \bigcup_n B_n$. This set need not be in $\mathcal{A}$, but we still have, because of (2),

$$c(B_\infty) \leq c(X \setminus (B_1 \cup \dots \cup B_m)) \leq 2v(B_{n+1}) \xrightarrow[n \to \infty]{} 0$$

since the series

$$\sum_{n\in\mathbb{N}} v(B_n) = \sup_N \sum_{n=1}^{N} v(B_n) = \sup_N v\left(\bigcup_{n=1}^{N} B_n\right) \le v(X)$$

converges. In particular, $\lim_{n\to\infty} v(X \setminus \bigcup_{i=1}^{n} B_i) = 0$.

Using again the convergence of the series $\sum_n v(B_n)$, we find some $N = N(\varepsilon)$ such that $\sum_{n>N} v(B_n) \le \varepsilon$, hence $B_1, B_2, \ldots, B_N$ and $X \setminus \bigcup_{n=1}^{N} B_n$ are the desired small pieces of $X$. This completes the proof that $v$ is sliceable.

We now show that sliceability implies condition (DD). Let $B \in \mathcal{A}$ with $v(B) > 0$. Since the "relative" finitely additive valuation $v_B(A) := v(A \cap B)/v(B)$ inherits the nonatomic property from $v$, it is clearly enough to show that for every $\alpha \in (0,1)$, there is an increasing sequence

$$B_\alpha^1 \subset B_\alpha^2 \subset B_\alpha^3 \subset \cdots, \quad B_\alpha^n \in \mathcal{A} \,:\, \sup_{n\in\mathbb{N}} v(B_\alpha^n) = \alpha,$$

which is the property (DD) relative to the full cake $X$ only.

Since $v$ is sliceable, there are mutually disjoint sets $C_1^n, \ldots, C_N^n \in \mathcal{A}$, where $N = N(n)$, $X = \bigcup_{i=1}^{N} C_i^n$, and $v(C_i^n) < \frac{1}{n}$.

Let $k = \lfloor 1/\alpha \rfloor + 1$. Set $B_k := C_1^k \cup \cdots \cup C_{M(k)}^k$, where $M(k) \in \{1, \ldots, N(k)\}$ is the unique number such that

$$\sum_{i=1}^{M(k)} v(C_i^k) \le \alpha < \sum_{i=1}^{M(k)+1} v(C_i^k) \le \sum_{i=1}^{M(k)} v(C_i^k) + \frac{1}{k}.$$

By construction, $\alpha \ge v(B_k) = \sum_{i=1}^{M(k)} v(C_i^k) > \alpha - \frac{1}{k}$. Thus, we can iterate this procedure, considering $X \setminus B_k$ and constructing a set $D_{k+1} \subseteq X \setminus B_k$ that satisfies

$$(\alpha - v(B_k)) \ge v(D_{k+1}) > (\alpha - v(B_k)) - \frac{1}{k+1}.$$

For $B_{k+1} := B_k \cup D_{k+1}$, we get $\alpha \ge v(B_{k+1}) > \alpha - \frac{1}{k+1}$.

The sequence $B_{k+i}$, $i \in \mathbb{N}$, satisfies $v(B_{k+i}) \uparrow \alpha$, i.e., $B_\alpha^n = B_{k+n}$ is the sequence of sets we need to have property (DD).

As mentioned earlier, (DD) implies atom-freeness, which completes this proof. □

Since for a $\sigma$-additive valuation on a $\sigma$-algebra $\mathcal{A}$, properties (D) and (DD) are equivalent, we immediately get:

*Corollary 12* Let $v$ be a $\sigma$-additive valuation on a $\sigma$-algebra $\mathcal{A}$ over an abstract cake $X$. The conditions (D), (DD), "$v$ is sliceable," and "$v$ has no atoms" are pairwise equivalent.

If $A_1, A_2, \ldots$ is an enumeration of the nonequivalent atoms of the $\sigma$-additive valuation $v$, then $A_\infty := X \setminus \bigcup_{n\in\mathbb{N}} A_n \in \mathcal{A}$, and we can restate Corollary 12 in the form of a decomposition theorem.

*Corollary 13* Let $v$ be a $\sigma$-additive valuation on a $\sigma$-algebra $\mathcal{A}$ over a(n abstract) cake $X$. Then $X$ can be written as a disjoint union of a $v$-sliceable set $A_\infty$ and at most countably many atoms $A_1, A_2, \ldots$.

# 3 Which Pieces Should Be Admissible?

In the cake-cutting literature, a great variety of different definitions have been used for the set $\mathcal{P}$ of admissible pieces of cake. We first collect the most commonly used definitions for $\mathcal{P}$, along with the corresponding references and discuss them in detail. Then we show several relations among these definitions and discuss what this implies for a most reasonable choice of $\mathcal{P}$.

Typical choices for the set $\mathcal{P}$ containing all admissible pieces of a standard cake $[0,1]$ are

1. all finite unions of intervals from $[0,1]$, i.e., the family $\mathcal{I}([0,1])$ defined earlier on page 8;
2. all countable unions of intervals from $[0,1]$, i.e., $\mathcal{I}([0,1])^{\mathbb{N}} = \left\{\bigcup_{i\in\mathbb{N}} I_i \mid I_i \in \mathcal{I}([0,1])\right\}$;
3. the Borel $\sigma$-algebra over $[0,1]$, i.e., $\mathcal{B}([0,1])$;
4. the set of all Lebesgue-measurable sets over $[0,1]$, i.e., $\mathcal{L}([0,1])$;[1] or
5. the power set $\mathfrak{P}([0,1])$ of $[0,1]$.

Assuming $\mathcal{P} = \mathcal{I}([0,1])$ is common among papers that consider only finite cake-cutting protocols. Such protocols can make only a finite number of cuts, thus producing a finite set of contiguous pieces, i.e., intervals, to be evaluated by the players. Authors that make this assumption and use $\mathcal{P} = \mathcal{I}([0,1])$ include Woeginger and Sgall (2007), Stromquist (2008), Lindner and Rothe (2009), Procaccia (2009), Walsh (2011), Cohler et al. (2011), Bei et al. (2012), Cechlárová and Pillárová (2012b), Brams et al. (2012), Cechlárová et al. (2013), Chen et al. (2013), Brânzei and Miltersen (2013), Aziz and Mackenzie (2016b,a, 2020), Edmonds and Pruhs (2006), and Aziz and Mackenzie (2016a).

As a special case, valuation functions may even be restricted to single intervals, which is done by Cechlárová and Pillárová (2012a) and Aumann and Dombb (2010). Even though the restriction to finite unions of intervals is sensible from a practical perspective, it may artificially constrain results that could hold also in a more general setting.

Brânzei et al. (2013) extend $\mathcal{P}$ to contain countably infinite unions of intervals, i.e., $\mathcal{I}([0,1])^{\mathbb{N}}$.

Authors assuming $\mathcal{P} = \mathcal{B}([0,1])$ include Stromquist and Woodall (1985), Deng et al. (2009), and Segal-Halevi et al. (2017).

Works using $\mathcal{P} = \mathcal{L}([0,1])$ include those by Reijnierse and Potters (1998), Arzi et al. (2011), and Robertson and Webb (1997). Additionally, several authors do not explicitly make the assumption $\mathcal{P} = \mathcal{L}([0,1])$, but they define valuation functions based on (Lebesgue-)measurable sets only, most prominently, a valuation function is often defined as the integral of a given probability density function on $[0,1]$. This or a similar assumption is made by Brams et al.

[1] Recall that a set $\tilde{B}$ is Lebesgue-measurable if, and only if, there is a Borel-measurable set $B$ such that the symmetric difference $\tilde{B} \mathbin{\triangle} B := (B \setminus B) \cup (B \setminus \tilde{B}) \subseteq N$ is contained in a Borel-measurable set $N$ with Lebesgue measure $\lambda(N) = 0$. We will see in Theorem 14 that there are indeed Lebesgue-measurable sets that are not Borel-measurable.

(2003, 2006, 2008, 2013), Robertson and Webb (1998), Webb (1997), Aumann et al. (2013), Brânzei et al. (2016), and Caragiannis et al. (2011).

Papers that assume $\mathcal{P} = \mathfrak{P}([0,1])$ include those by Maccheroni and Marinacci (2003), Sgall and Woeginger (2007), Saberi and Wang (2009), Manabe and Okamoto (2010), and Aumann et al. (2014).

Finally, several works, including those by Dubins and Spanier (1961), Barbanel (1996a,b), Zeng (2000), and Brams and Taylor (1995b), define the set of admissible pieces of cake to be some ($\sigma$-)algebra (not necessarily Borel) over $[0,1]$.

Note that each of the sets $\mathcal{I}([0,1])$, $\mathcal{B}([0,1])$, $\mathcal{L}([0,1])$, and $\mathfrak{P}([0,1])$ is an algebra over $[0,1]$, and all, except $\mathcal{I}([0,1])$, are also $\sigma$-algebras over $[0,1]$. However, $\mathcal{I}([0,1])^{\mathbb{N}}$ is not an algebra, as the proof of the following theorem shows.

Having introduced all the different approaches currently used in the literature, we will now prove the strict inclusions among these sets stated in the following theorem.

**Theorem 14** $\mathcal{I}([0,1]) \overset{\text{(a)}}{\subsetneq} \mathcal{I}([0,1])^{\mathbb{N}} \overset{\text{(b)}}{\subsetneq} \mathcal{B}([0,1]) \overset{\text{(c)}}{\subsetneq} \mathcal{L}([0,1]) \overset{\text{(d)}}{\subsetneq} \mathfrak{P}([0,1])$.

*Proof* We start with proving (a): $\mathcal{I}([0,1]) \subsetneq \mathcal{I}([0,1])^{\mathbb{N}}$. Obviously, $\mathcal{I}([0,1]) \subseteq \mathcal{I}([0,1])^{\mathbb{N}}$ is true, as every finite union of intervals is a countable union of intervals. To see that the two sets are not equal, look at $I = \bigcup_{i \in \mathbb{N} \cup \{0\}} [3 \cdot 2^{-i-2}, 2^{-i}]$. It is clear that $I \in \mathcal{I}([0,1])^{\mathbb{N}}$ is true, as $I$ is a countable union of intervals. However, it holds that $I = [3/4, 1] \cup [3/8, 1/2] \cup \cdots$, i.e., $I$ cannot be written as a finite union of intervals, as all these subintervals are pairwise disjoint. Hence, $I \notin \mathcal{I}([0,1])$, so $\mathcal{I}([0,1]) \subsetneq \mathcal{I}([0,1])^{\mathbb{N}}$, and we have shown (a).

In Lemma 9 and Definition 6, we have seen that $\mathcal{B}([0,1]) = \sigma(\mathcal{Q}([0,1]))$ where $\mathcal{Q}([0,1])$ is the family of all intervals within $[0,1]$. Since a $\sigma$-algebra is stable under (finite and countable) unions, we get $\mathcal{I}([0,1]) \subseteq \sigma(\mathcal{Q}([0,1])) = \mathcal{B}([0,1])$. Using again the stability of a $\sigma$-algebra under countable unions, we arrive at $\mathcal{I}([0,1])^{\mathbb{N}} \subseteq \mathcal{B}([0,1])$.

Since, however, $\mathbb{Q} \cap [0,1] \in \mathcal{B}([0,1])$ is true, as $\mathbb{Q} \cap [0,1]$ can be written as a countable union of intervals that each contain one element, it must hold that $\overline{\mathbb{Q} \cap [0,1]} \in \mathcal{B}([0,1])$ by the definition of a $\sigma$-algebra. However, the irrational numbers $\overline{\mathbb{Q} \cap [0,1]}$ in $[0,1]$ cannot be written as a countable union of intervals, since every interval containing more than one element immediately contains a rational number. Therefore, $\mathcal{I}([0,1])^{\mathbb{N}}$ is not an algebra and $\mathcal{I}([0,1])^{\mathbb{N}} \neq \mathcal{B}([0,1])$ holds, proving (b).

The inclusion $\mathcal{B}([0,1]) \subseteq \mathcal{L}([0,1])$ holds by definition, as all Borel sets are Lebesgue-measurable. However, there are Lebesgue-measurable sets that are not Borel-measurable: Observe that the cardinality of $\mathcal{L}([0,1])$ is the cardinality of $\mathfrak{P}([0,1])$ (which is $2^{\mathfrak{c}} > \mathfrak{c}$), whereas there are only continuum-many (i.e., $\mathfrak{c}$, the cardinality of $[0,1]$) Borel sets (see Schilling, 2017, Appendix G, Corollary G.7). This proves (c). An alternative direct construction can be based on the Cantor function, also known as the *devil's staircase* (see Schilling and Kühn, 2021, p. 153, Example 7.20).

Finally, the power set $\mathfrak{P}([0,1])$ trivially contains all other families of sets considered earlier. Nevertheless, there are sets in $\mathfrak{P}([0,1])$ that are not Lebesgue-measurable, for example the Vitali set that we introduced in Example 2, so $\mathcal{L}([0,1]) \neq \mathfrak{P}([0,1])$, and we have (d). □

# 4 Discussion

Taking $\mathcal{P} = \mathcal{I}([0,1])$ as domain for a valuation $v$ and a protocol involving a finite number of cuts is always possible. If we are open-ended or even infinite, the naive choice $\mathcal{P} = \mathcal{I}([0,1])^{\mathbb{N}}$ is problematic, as $\mathcal{I}([0,1])^{\mathbb{N}}$ is not an algebra and thus does not even satisfy the minimum requirements for $\mathcal{P}$ as described in the first paragraph of Section 2.1.

From a theoretical point of view, however, the choice $\mathcal{P} = \mathcal{I}([0,1])$ may be unnecessarily restrictive, especially in light of the fact that we also want to use infinite cake-cutting protocols. Therefore, a larger set $\mathcal{P}$ may be desirable, perhaps even larger than $\mathcal{I}([0,1])^{\mathbb{N}}$, which (as we have seen) has disqualified itself.

We start our discussion by explicating why $\mathcal{P} = \mathfrak{P}([0,1])$ is a bad choice and we then provide arguments for a better option, namely the Borel $\sigma$-algebra $\mathcal{P} = \mathcal{B}([0,1])$.

## 4.1 Taming $\mathfrak{P}([0,1])$ with Exotic Contents via Banach Limits

If one boldly desires to define valuation functions on the set $\mathfrak{P}([0,1])$ of all subsets of the cake, it remains to be shown that this indeed is possible. We have seen that the commonly used valuation functions represented via boxes, as depicted in Figure 2, are not capable of evaluating every piece of cake in $\mathfrak{P}([0,1])$. Hence, in this section we aim to define a valuation function capable of evaluating *every* possible piece of cake in $\mathfrak{P}([0,1])$.

Let us begin with a negative result.

### 4.1.1 A Negative Result

Using axiomatic set theory one can show that there cannot be a valuation $v$ of the standard cake $[0,1]$ which

- is defined on all of $\mathfrak{P}([0,1])$,
- is $\sigma$-additive, and
- is divisible, hence satisfies $v(\{x\}) = 0$ for any $x \in [0,1]$.

This is the consequence of a result by Ulam, and it requires that the continuum hypothesis holds true, see the books by Oxtoby (1980, p. 26, Proposition 5.7) or Schilling and Kühn (2021, pp. 132–3, Example 6.15).

This means that we should look for finitely additive valuations if we want to admit all pieces of cake. Let us formally define a valuation function $\mu$ on $\mathfrak{P}([0,1])$ satisfying the requirements (M), (A), and (D) from Definition 2. To

do so, in a first step, we must choose an arbitrary sequence $(x_i)_{i\in\mathbb{N}}$ of pairwise distinct elements from $[0,1]$. For every $A \subseteq [0,1]$, we define a mapping $f_A\colon \mathbb{N} \to [0,1]$ with

$$n \mapsto f_A(n) = \frac{|A \cap \{x_1, \dots, x_n\}|}{n},$$

where $|B|$ denotes the cardinality of any set $B$. That is, $f_A(n)$ describes the relative frequency of the first $n$ elements of $(x_i)_{i\in\mathbb{N}}$ being in $A$. For some sets $A$ the limit $\lim_{n\to\infty} f_A(n)$ does exist, but it may not exist for other sets $A$. We can, however, use the Banach limits, that we will now introduce.

### 4.1.2 Banach Limits

We will need a nonconstructive way to extend linear maps. The key result is the standard Hahn–Banach theorem, which is well-known from functional analysis (see, e.g., Rudin, 1991, Theorem 3.2), so we need to go on a quick excursion into functional analysis.

**Theorem 15** *Assume that $(Y, \|\cdot\|)$ is a normed vector space and $L : M \to \mathbb{R}$ a linear functional, which is defined on a linear subspace $M \subseteq Y$ satisfying $|Lx| \le \kappa\|x\|$ for all $x \in M$ with a universal constant $\kappa = \kappa_L \in (0,\infty)$. Then there is an extension $\hat{L} : Y \to \mathbb{R}$ such that $\hat{L}$ is again linear and satisfies $|\hat{L}x| \le \kappa\|x\|$ for all $x \in Y$ with the same constant $\kappa = \kappa_L$ as before.*

With a little more effort, but essentially the same proof, we can replace the norm $\|x\|$ (respectively, $\kappa\|x\|$) by a general sublinear map $p : Y \to \mathbb{R}$. Sublinear means that $p(\alpha x) = \alpha p(x)$ and $p(x+y) \le p(x)+p(y)$ for all $x, y \in Y$ and $\alpha \ge 0$. In this case, the extension of $Lx \le p(x)$ satisfies $-p(-x) \le \hat{L}x \le p(x)$. Note that $p$ is only positively homogeneous, i.e., it may happen that $-p(-x) \ne p(x)$.

The proof is nonconstructive and, at least for nonseparable spaces $Y$, relies on the axiom of choice.

We will use the Hahn–Banach theorem for the space of bounded sequences $\ell^\infty([0,\infty)) = \{x = (x_n)_{n\in\mathbb{N}} \subset [0,\infty) \mid \|x\|_\infty < \infty\}$, where $\|x\|_\infty = \sup_{n\in\mathbb{N}} x_n$ is the uniform norm. Note that $(\ell^\infty([0,\infty)), \|\cdot\|_\infty)$ is a nonseparable space.

A prime example of a bounded linear functional is the limit: Consider those $x = (x_n)_{n\in\mathbb{N}} \in \ell^\infty([0,\infty))$ where $L(x) := \lim_{n\to\infty} x_n = x$ exists in the usual sense. It is common to write $c([0,\infty)) = \{x \in \ell^\infty([0,\infty)) \mid \lim_{n\to\infty} x_n \text{ exists}\}$. Clearly, $\lim_{n\to\infty} x_n = \limsup_{n\to\infty} x_n \le \sup_{n\in\mathbb{N}} x_n$, so that $L$ is a bounded linear functional on $M = c([0,\infty)) \subset Y = \ell^\infty([0,\infty))$, and we can extend it to all of $Y$ as the **Banach limit**, i.e.,

$$\operatorname*{LIM}_{n\to\infty} x_n := \begin{cases} \lim_{n\to\infty} x_n & \text{if } x \in c([0,\infty)), \\ \hat{L}(x) & \text{if } x \in \ell^\infty([0,\infty)) \setminus c([0,\infty)). \end{cases}$$

Using the addition to the Hahn–Banach theorem with $p(x) := \limsup_{n\to\infty} x_n$ and the observation that $\lim_{n\to\infty} x_n$ exists if, and only if, $\liminf_{n\to\infty} x_n = \limsup_{n\to\infty} x_n \in [0,\infty)$, we can choose the extension $\hat{L}$ in such a way that

$$\liminf_{n\to\infty} x_n \leq \operatorname*{LIM}_{n\to\infty} x_n \leq \limsup_{n\to\infty} x_n.$$

The construction of Banach limits is a typical application of the Hahn–Banach extension theorem, hence the axiom of choice. The appearance of these two concepts in this context is not an accident. The seminal paper of Banach (1923) (see also Banach, 1932, Chapter II.§1) proves what we now call the "Hahn–Banach extension theorem for linear functionals" in order to solve the *problème de la mésure* by Lebesgue (1904, Chapter VII.ii) which asks for the existence of an additive, or $\sigma$-additive, translation invariant measure on $\mathfrak{P}(\mathbb{R}^n)$. The answer depends on the dimension: In dimension $n \geq 3$, it is always negative (because of the Banach–Tarski paradox), whereas in dimensions 1 and 2 it is negative if the measure is to be $\sigma$-additive (because of Vitali-type constructions, cf. Example 2). More on this can be found in the books by Wagon (1985, Chapter 10) and Schilling and Kühn (2021, Example 7.31).

There is a deep connection between the underlying group structure of the space $\mathbb{R}^n$ and Lebesgue's measure problem (this was discovered by von Neumann, 1929). Following M. M. Day, a group $\mathbb{G}$ which allows for finitely additive, (left-)translation invariant measures on all of $\mathfrak{P}(\mathbb{G})$ is nowadays called **amenable** – a pun combining the actual meaning of the word ("nice, comfortable") with its pronunciation which reminds of "mean value" or measure. The axiom of choice, which is needed for Hahn–Banach, can also be used to construct extensions of measures defined on a sub-algebra $\mathcal{A}_0$ of an algebra $\mathcal{A}$. It is known that this extendability, essentially, is equivalent to the Hahn–Banach theorem (cf. Wagon, 1985, Theorem 10.11 and Corollary 13.6) describing its axiomatic strength.

### 4.1.3 From Banach Limits to Valuation Functions

Having defined and discussed Banach limits, we will now use them to construct a valuation function

$$\mu\colon \mathfrak{P}([0,1]) \to [0,1], \quad A \mapsto \operatorname*{LIM}_{n\to\infty} f_A(n).$$

It is clear that $\mu(A)$ is additive since both the limit and the Banach limit are additive, so property (A) from Definition 2 is satisfied. In the following lemma we show that $\mu$ satisfies property (D). At first glance, this seems to contradict Corollary 8. But divisibility (D) involves the domain of the valuation, and the proof of the lemma shows that we have almost no control on the set $A_\alpha \subseteq A$ which achieves divisibility. That means, the following phenomenon is symptomatic for having a "too big domain."

*Lemma 16* For every $A \in \mathfrak{P}([0,1])$ with $\mu(A) > 0$ and every real number $\alpha \in [0,1]$, there exists a subset $A_\alpha \subseteq A$ in $\mathfrak{P}([0,1])$ such that $\mu(A_\alpha) = \alpha\mu(A)$.

*Proof* If $\mu(A) > 0$ then $A$ must contain an infinite number of points of the underlying sequence, say $A \cap \{x_1, x_2, \dots\} = \{x_{i(1)}, x_{i(2)}, \dots\}$ for some increasing sequence $(i(k))_{k\in\mathbb{N}}$ of integers. By assumption, $A \cap \{x_1, x_2, \dots, x_n\} = \{x_{i(1)}, \dots, x_{i(m)} \mid i(m) \leq n\}$, and so $\mu(A) = \mathrm{LIM}_{n\to\infty} \frac{1}{n}|\{x_{i(1)}, \dots, x_{i(m)} \mid i(m) \leq n\}|$. We have to construct a set $B \in \mathfrak{P}(A)$ such that $\mathrm{LIM}_{n\to\infty} f_B(n) = \alpha\mu(A)$ for fixed $\alpha \in [0,1]$.

The key observation in this proof is the fact that for any nonnegative rational number $k/n$ with $k < n$, we have

$$\frac{k}{n+1} < \frac{k}{n} < \frac{k+1}{n+1},$$

i.e., the quantity $f_B(n) = \frac{1}{n}|B \cap \{x_{i(1)}, \dots, x_{i(m)} \mid i(m) \leq n\}|$ decreases if we jack up $n \to n+1$ and the numerator does not increase, i.e., if $i(m+1) > n+1$ or if $x_{i(m+1)} = x_{n+1} \notin B$, and it increases if we jack up $n \to n+1$ and $x_{i(m+1)} = x_{n+1} \in B$.

Fix $\alpha \in [0,1]$ and observe that we can assume that $0 < \alpha < 1$: If $\alpha = 0$, we take $B = \emptyset$, and for $\alpha = 1$, we use $B = \{x_{i(m)} \mid m \in \mathbb{N}\}$. For $\alpha \in (0,1)$, we use a recursive approach.

Since $\mu(A) = \mathrm{LIM}_{n\to\infty} \frac{1}{n}|\{x_{i(1)}, \dots, x_{i(m)} \mid i(m) \leq n\}|$ and $0 < \alpha < 1$ we must have $\frac{1}{n(1)}|\{x_{i(1)}, \dots, x_{i(m)} \mid i(m) \leq n(1)\}| \geq \alpha\mu(A)$ for some $n(1) \in \mathbb{N}$. Define $B_{n(1)} = \{x_{i(1)}, \dots, x_{i(m)} \mid i(m) \leq n(1)\}$, and assume that we have already found a set $B_n$ such that $f_{B_n}(n) \geq \alpha\mu(A)$. Because of the observation at the beginning of the proof, the numbers

$$\ell_{n+1} := \min\{k > n \mid f_{B_n}(k) \leq \alpha\mu(A)\} \text{ and}$$
$$u_{n+1} := \min\left\{k > \ell_{n+1} \mid f_{A_k}(k) \geq \alpha\mu(A) \text{ for } A_k = B_n \cup \{x_{i(\ell_{n+1}+1)}, \dots, x_{i(k)}\}\right\}$$

are well-defined and satisfy $\ell_n < u_n < \ell_{n+1} < u_{n+1}$ and $\ell_{n+1} \to \infty$. Setting

$$B_{n+1} = B_n \cup \{x_{i(\ell_{n+1}+1)}, \dots, x_{i(u_{n+1})}\}$$

finishes the recursion, and we can define $B = \bigcup_{n\geq n(1)} B_n$.

By construction, $|f_B(n) - \alpha\mu(A)| \leq \ell_{n+1}^{-1}$ holds for $n > n(1)$, completing this proof. □

We now provide a counterexample that shows that $\mu$ is not $\sigma$-additive. To do so, we define $A_0 = [0,1] \setminus \{x_i \mid i \in \mathbb{N}\}$ and $A_i = \{x_i\}$ for $i \in \mathbb{N}$. Obviously, for all $j \in \mathbb{N} \cup \{0\}$, it holds that $\mu(A_j) = 0$, while at the same time we have

$$\mu\left(\bigcup_{j\in\mathbb{N}\cup\{0\}} A_j\right) = \mu([0,1]) = 1,$$

which means that $\mu$ is not $\sigma$-additive.

The valuation function $\mu$ defined above may seem to be attractive for cake-cutting. We can interpret the sequence $(x_i)_{i\in\mathbb{N}}$ as countably many points which are used to evaluate arbitrary pieces of the cake. However, there are multiple

drawbacks. First of all, the existence of a Banach limit is only guaranteed if one is willing to accept the validity of the axiom of choice, as already mentioned in Section 4.1.2. Furthermore, until now no explicit nontrivial example of a Banach limit is known. Hence, we cannot calculate $\mu(A)$ for $A \in \mathfrak{P}([0,1])$ if the ordinary limit of $f_A(n)$ does not exist, as we do not know what the Banach limit looks like.

Thus, although $\mu$ is theoretically capable of evaluating all pieces of cake in $\mathcal{P} = \mathfrak{P}([0,1])$, it is actually not useful for our purposes. Besides the previously listed mathematical problems, there are also practical problems related to cake-cutting itself. If we would use $\mu$ as a valid valuation function in cake-cutting, all players would be obliged to precisely define a countable sequence $(x_i)_{i\in\mathbb{N}}$ of pairwise distinct elements in $[0,1]$ and some Banach limit they are using for their valuation functions. When we think of common approaches and results in the cake-cutting literature, this approach seems impractical and not feasible to use.

Hence, finding practically usable valuation functions defined on $\mathfrak{P}([0,1])$ seems to remain an open problem. Nonetheless, this section showed that defining more complex valuation functions (compared to the valuation functions represented via boxes) does not solve our initial problem on $\mathfrak{P}([0,1])$. Therefore, in the next section we discuss an alternative solution, namely, reducing $\mathcal{P}$ in size from $\mathfrak{P}([0,1])$ to a smaller subfamily contained in $\mathfrak{P}([0,1])$.

## 4.2 Borel $\sigma$-Algebra

We recommend to use $\mathcal{P} = \mathcal{B}([0,1])$ as the most useful family of all admissible pieces of cake. As shown in Theorem 14, the Borel $\sigma$-algebra $\mathcal{B}([0,1])$ (strictly) contains $\mathcal{I}([0,1])$ as well as $\mathcal{I}([0,1])^{\mathbb{N}}$, but is strictly smaller than $\mathcal{L}([0,1])$ and $\mathfrak{P}([0,1])$.

In general, the Borel $\sigma$-algebra can become quite large and complicated if the base set is not countable, as is the case for $[0,1] \subset \mathbb{R}$. In particular, one needs transfinite induction to "construct" all Borel sets. This means that, in general, we cannot construct a valuation $\mu$ on $\mathcal{B}([0,1])$ by explicitly assigning a value $\mu(A)$ to every element $A \in \mathcal{B}([0,1])$ nor give a recursive algorithm to construct $\mu(A)$, as the $\sigma$-algebra is simply too large. Instead, one can describe the valuation on a suitable generator of the $\sigma$-algebra and use Carathéodory's extension theorem, stated previously as Theorem 10.

Let us show here that the box-based valuation functions are $\sigma$-additive valuations on $\mathcal{I}([0,1])$ and that $\mathcal{I}([0,1])$ is an algebra. In this case, we can use Carathéodory's extension theorem to extend the valuation functions to measures on $\sigma(\mathcal{I}([0,1])) = \mathcal{B}([0,1])$. Since the valuation functions are $\sigma$-finite, it follows that this extension is unique. Hence, by providing a box-based valuation function, we obtain a unique measure on $\mathcal{B}([0,1])$. Thus $\mathcal{P} = \mathcal{B}([0,1])$ is a good solution to our problem.

Let us formalize the box-based valuation functions. A box-based valuation function $\mu$ partitions the complete cake $[0,1]$ into a finite number of pairwise disjoint subintervals, where each subinterval is allocated a finite number of

boxes of equal height. We denote the set of all subintervals which $\mu$ uses by

$$\mathcal{I}_\mu = \{I_1 = [a_1, b_1), \ldots, I_{n-1} = [a_{n-1}, b_{n-1}), I_n = [a_n, b_n]\},$$

where $\bigcup_{i=1}^n I_i = [0,1]$ and we have $I_i \cap I_j = \emptyset$ for all $i$ and $j$, $1 \leq i < j \leq n$. Furthermore, denote by $\psi_i \in \mathbb{N}$, for $1 \leq i \leq n$, the number of boxes allocated to an interval $I_i$ by $\mu$ and denote by $\psi_\mu = \sum_{i=1}^n \psi_i$ the total number of boxes. This gives the following weight function

$$p(x) := \frac{1}{\psi_\mu} \sum_{i=1}^n \frac{\psi_i}{\lambda(I_i)} \mathbf{1}_{I_i}(x) = \frac{1}{\psi_\mu} \sum_{i=1}^n \frac{\psi_i}{b_i - a_i} \mathbf{1}_{I_i}(x).$$

Note that $p(x)$ is a Borel-measurable function, which is a probability density, i.e., $\int_0^1 p(x)\,dx = 1$. Since $\mu$ is a Lebesgue measure with a weight, we cannot define it on all of $\mathfrak{P}([0,1])$, but we may extend it easily onto $\mathcal{B}([0,1])$ using integration: Define $\mu\colon \mathcal{B}([0,1]) \to [0,1]$ as

$$B \mapsto \mu(B) := \int_B p(x)\,dx.$$

In particular, if $B \in \mathcal{I}([0,1])$ is a finite union of intervals in $[0,1]$, we see that

$$B \cap I_i = \bigcup_{j=1}^{n(B,i)} \langle c_j^i, d_j^i \rangle, \quad i = 1, 2, \ldots n$$

for suitable $n(B,i) \in \mathbb{N}$, and

$$\mu(B) = \frac{1}{\psi_\mu} \sum_{i=1}^n \left[ \frac{\psi_i}{b_i - a_i} \sum_{j=1}^{n(B,i)} (d_j^i - c_j^i) \right].$$

*Example 17* Referring back to the box-based valuation function $\nu$ from Figure 2 on page 4, we obtain

$$\mathcal{I}_\nu = \{I_1 = [0, 1/6), I_2 = [1/6, 2/6), \ldots, I_6 = [5/6, 1]\}.$$

Also, we have $\psi_1 = 2$, $\psi_2 = 1$, $\psi_3 = 5$, $\psi_4 = 2$, $\psi_5 = 4$, $\psi_6 = 3$, and $\psi_\nu = 17$. For $B = [0, 2/6]$, we obtain

$$\begin{aligned}
\nu(B) &= \frac{1}{\psi_\nu} \sum_{i=1}^6 \left[ \frac{\psi_i}{b_i - a_i} \sum_{j=1}^{n(B,i)} (d_j^i - c_j^i) \right] \\
&= \frac{1}{17} \left( \frac{2}{1/6} \cdot (1/6 - 0) + \frac{1}{1/6} \cdot (2/6 - 1/6) + \frac{5}{1/6} \cdot 0 + \frac{2}{1/6} \cdot 0 + \frac{4}{1/6} \cdot 0 + \frac{3}{1/6} \cdot 0 \right) \\
&= \frac{3}{17}.
\end{aligned}$$

Summing up, $\mathcal{B}([0,1])$ is recommended as a very good choice for $\mathcal{P}$, since this choice enables us to use box-based valuation functions and their extensions as measures. It also enables us to use any Borel-measurable probability density – not only piecewise continuous densities – to define a divisible valuation function on $\mathcal{B}([0,1])$ which is an absolutely continuous probability measure with respect to Lebesgue measure. In this case, a further extension to $\mathcal{L}([0,1])$ is also possible, but the enrichment by subsets of Borel null sets (which are evaluated zero) has no additional benefit.

# 5 Conclusion and Some Further Technical Remarks

Among the questions we have tried to answer are:

1. Which subsets of $[0,1]$ should be considered as pieces of cake? Only finite unions of intervals or more general sets?
2. If valuation functions are considered as set-functions as studied in measure theory, should they be *$\sigma$-additive* or only *finitely additive*?

A related interesting question is:

3. Which continuity property should be used for a valuation?

   For the standard cake $[0,1]$, the natural choices are either *divisibility* (D) or *absolute continuity* with respect to Lebesgue measure, see p. 14. Obviously, absolute continuity implies continuity. There is a partial converse to this assertion: The notions of continuity and divisibility coincide (cf. Corollary 8) and the distribution function $F_v(x)$ of a continuous valuation can be represented as a sum of the form $F_v(x) = \int_0^x f(t)\,dt + v^{\mathrm{sc}}([0,x])$; this means that it has an absolutely continuous part and a continuous-singular part, see the discussion in the paragraph on "General Measures" following Definition 7. For an abstract cake, one should replace divisibility (D) by the notion of sliceability, which is equivalent to condition (DD) by Theorem 11, see Section 2.4 and Schilling and Stoyan (2016).

While one can define the Dirac and counting measures for all sets in $\mathfrak{P}(X)$, there is no way to define a geometrically sensible (and $\sigma$-additive [in dimensions one and two] or finitely additive [in all higher dimensions]) notion of "volume" for all sets – if we accept the validity of the axiom of choice. One can even show that the axiom of choice is equivalent to the existence of non-measurable sets (cf. Ciesielski, 1989, p. 55).

Our findings result in concrete recommendations for cake-cutters. For a finitely additive valuation $v$ on the standard cake $[0,1]$ (or indeed any one-dimensional cake) equipped with the algebra $\mathcal{I}([0,1])$ generated by the intervals, divisibility (D) is equivalent to atom-freeness or the continuity of the distribution function $F_v(x) = v([0,x])$, cf. Corollary 8. For an abstract cake

and a finitely additive valuation $v$, divisibility (D) should be replaced by sliceability (DD), which is equivalent to $v$ being atom-free; if $v$ is even $\sigma$-additive, conditions (D) and (DD) coincide, see Theorem 11 and Corollary 12.

All of this breaks down, however, if we consider finitely additive valuations on too big domains, say $\mathcal{P} = \mathfrak{P}(X)$: Even for the standard cake there are divisible, finitely additive but not $\sigma$-additive valuations, see Lemma 16.

We have also discussed in detail the measure-theoretic notions and results that are relevant for the foundations of cake-cutting, for both the standard cake and abstract cakes, including the notions of $\sigma$-additivity, the Borel $\sigma$-algebra, and Carathéodory's extension theorem (Theorem 10). We emphasized the importance of the Hahn–Banach theorem and the underlying axiom of choice if one needs to evaluate arbitrary pieces of cake which are not Borel or Lebesgue sets.

Banach, who can be seen as one of the founding fathers of the field of cake-cutting,[2] might perhaps have appreciated the close connection between his work in measure theory and in cake-cutting. For future work, we suggest to study which implications our findings may have on existing or on yet-to-be-designed cake-cutting algorithms.

To conclude, we have surveyed the existing rich literature on cake-cutting algorithms and have identified the most commonly used choices of sets consisting of what is allowed as pieces of cake. After showing that these five most commonly used sets are distinct from each other, we have discussed them in comparison. In particular, we have argued that $\mathfrak{P}(X)$ is too general to define a (practically or theoretically) useful valuation function on it. And finally, we have reasoned why we recommend the Borel $\sigma$-algebra $\mathcal{B}(X)$ as a very good choice and how to construct, using Carathéodory's extension theorem, a measure on $\mathcal{B}(X)$ that cake-cutters can use to handle their box-based and even more general valuation functions.

For a pragmatic approach to cake-cutting on the standard cake $[0, 1]$, the following five points are important:

1. If one is interested in a fixed number of players and a fixed number of cuts, any additive valuation $v$ defined on the algebra of intervals $\mathcal{I}([0, 1])$ will do.
2. If the players take rounds and if the protocol is open-ended or infinite,[3] the finite additivity of the valuation $v$ needs to be strengthened to $\sigma$-additivity, and the domain of the valuation should contain the Borel $\sigma$-algebra $\mathcal{B}([0, 1])$ – this is the smallest $\sigma$-algebra containing $\mathcal{I}([0, 1])$.

[2]Indeed, Steinhaus (1948) presents the so-called last-diminisher procedure that is due to his students Banach and Knaster and guarantees a proportional division of the cake among any number of players.

[3]For example, prior to the celebrated finite bounded envy-free cake-cutting protocol due to Aziz and Mackenzie (2016a, 2020), the cake-cutting protocol of Brams and Taylor (1995a) was the best protocol known to guarantee envy-freeness for any number of players. They argue that the allocation must become envy-free at some (unknown) finite stage, which is why their protocol is considered to be a *finite unbounded* envy-free procedure only. And yet, being open-ended, it is in some sense even an *infinite* procedure that describes an infinite process. Similarly, it is reasonable to conjecture that some moving-knife procedures can be converted to discrete procedures that require infinitely many cuts.

3. If the valuation $v$ on $\mathcal{I}([0,1])$ is divisible, measure theory guarantees that one is automatically in the situation described in item 2, i.e., the proper domain of (the extension of) $v$ is the Borel $\sigma$-algebra $\mathcal{B}([0,1])$.
4. If one wants to extend the domain of the valuation $v$ beyond $\mathcal{B}([0,1])$, things become difficult: On the one hand, it is quite tricky to "construct" sensible valuations – unless we are happy with "rather simple" valuations like countable sums of point masses $v = \sum_{i\in\mathbb{N}} p_i \delta_{x_i}$, $\sum_{i\in\mathbb{N}} p_i = 1$, $(x_i)_{i\in\mathbb{N}} \subseteq [0,1]$, but these are obviously not divisible – and, on the other hand, they are not well-behaved, touching the very basis of axiomatic set theory.
5. The tools provided by measure theory are powerful enough to handle even abstract cakes.

**Acknowledgments.** We thank William S. Zwicker for helpful discussions. This work was supported in part by grants of the Deutsche Forschungsgemeinschaft (DFG) RO-1202/14-2, RO-1202/21-1 and the DFG and Narodowe Centrum Nauki (NCN) joint initiative "Beethoven 3 classic" SCHI-419/11-1, NCN 2018/31/G/ST1/02252. The second author was a member of the PhD-programme "Online Participation," supported by the North Rhine-Westphalian funding scheme "Forschungskollegs."

# Appendix A An Alternative Example Illustrating the Cantor Dust

*Example 18* (Cantor dust; Cantor's ternary set) Write the elements $x \in [0,1]$ of the cake $[0,1]$ as ternary numbers, i.e., in the form

$$x = \sum_{n\in\mathbb{N}} \frac{x_n}{3^n} \simeq 0.x_1x_2x_3\ldots, \quad \text{where} \quad x_n \in \{0,1,2\},$$

and consider the set $C_{1/3}$ comprising all $x$ whose ternary expansion contains the digits "0" or "2" only. To enforce uniqueness, identify expressions of the form $0.{*}{*}{*}1000\ldots$ with $0.{*}{*}{*}0222\ldots$. The set $C_{1/3}$ is not countable since there is a bijection between $C_{1/3}$ and $[0,1]$: Take any $x = 0.x_1x_2x_3\ldots \in C_{1/3}$ and read $\hat{x} := 0.\frac{x_1}{2}\frac{x_2}{2}\frac{x_3}{2}\ldots$ as **dyadic** expansion of an arbitrary element $\hat{x} \in [0,1]$.

The set $C_{1/3}$ is the so-called **Cantor set** from Example 1. Think of its elements as "cream" pieces within the cake $[0,1]$, and imagine two players, taking turns in picking pieces of cake; for some reason (that their cardiologist elaborated on in detail) they have to avoid the cream altogether. For this, they are allowed to make two cuts, taking out an interval from the cake.[4]

The optimal strategy is to take, in each round, the largest (necessarily open) interval between two cream pieces. From the triadic expansion, we see that, at each stage of the game, the maximum distance between two cream pieces is $0.\underbrace{{*}{*}{*}2}_{n}000\ldots - 0.\underbrace{{*}{*}{*}0}_{n}222\ldots = 0.\underbrace{0001}_{n}000\ldots \simeq 3^{-n}$, and this situation appears exactly $2^{n-1}$ times, since we have $2^{n-1}$ choices for the leading $n-1$ digits denoted by the wildcard "$*$" – to wit, the pieces taken out are always the middle thirds of the largest remaining interval of cake:

$$A_0 = [0,1] \xrightarrow{(*)} A_1 = A_0 \setminus (1/3, 2/3) = [0, 1/3] \cup [2/3, 1]$$

[4] This is, of course, a non-standard cake-cutting protocol.

$$\xrightarrow{(**)} A_2 = [0, 1/9] \cup [2/9, 1/3] \cup [2/3, 7/9] \cup [8/9, 1].$$

At the step marked $(*)$ Player 1 takes the first middle third, at the (double) step marked $(**)$ Player 2 and then Player 1 take the middle thirds of the remaining intervals, etc.

If this procedure is repeated on and on, we remove countably many intervals from $[0, 1]$ and end up with the **Cantor (ternary) set** $C_{1/3} = \bigcap_{n\in\mathbb{N}} A_n$ from Example 1, see Figure 1.

We can use the triadic expansion also to assign a unique code to the removed piece: $I_{t_1 t_2 \ldots t_n 2}$ denotes the newly removed piece of cake at stage $n + 1$ and the $t_1, \ldots, t_n \in \{0, 2\}$ mark the right-end point of the interval using the triadic expansion: $\sup I_{t_1 t_2 \ldots t_n 2} = \sum_1^n t_i 3^{-i} + 2 \cdot 3^{-n-1}$. This allows us to come up with a formula for the Cantor function from Example 3:

$$\text{on all of } I_{t_1 t_2 \ldots t_n 2} \text{ the function } V \text{ has the value } \sum_{i=1}^{n} \frac{t_i}{2} 2^{-i} + 3^{-n-1}. \tag{A1}$$

We refer to (Schilling and Kühn, 2021, Sec. 2.5, 2.6) for a full discussion of this.

# References

Arzi, O., Y. Aumann, and Y. Dombb. 2011. Throw one's cake—and eat it too, *Proceedings of the 2nd International Conference on Algorithmic Decision Theory*, 69–80. Springer.

Aumann, Y. and Y. Dombb. 2010, December. The efficiency of fair division with connected pieces, *Proceedings of the 6th International Workshop on Internet & Network Economics*, 26–37. Springer-Verlag *Lecture Notes in Computer Science #6484*.

Aumann, Y., Y. Dombb, and A. Hassidim 2013. Computing socially-efficient cake divisions. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems*, pp. 343–350. International Foundation for Autonomous Agents and Multiagent Systems.

Aumann, Y., Y. Dombb, and A. Hassidim 2014. Auctioning a cake: Truthful auctions of heterogeneous divisible goods. In *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems*, pp. 1045–1052. International Foundation for Autonomous Agents and Multiagent Systems.

Aziz, H. and S. Mackenzie 2016a, October. A discrete and bounded envy-free cake cutting protocol for any number of agents. In *Proceedings of the 57th IEEE Symposium on Foundations of Computer Science*, pp. 416–427. IEEE Computer Society Press.

Aziz, H. and S. Mackenzie 2016b, June. A discrete and bounded envy-free cake cutting protocol for four agents. In *Proceedings of the 48th Annual ACM Symposium on Theory of Computing*, pp. 454–464. ACM Press.

Aziz, H. and S. Mackenzie. 2020. A bounded and envy-free cake cutting algorithm. *Communications of the ACM 63*(4): 119–126 .

Banach, S. 1923. Sur le problème de la mésure. *Fundamenta Mathematica* 4: 7–33 .

Banach, S. 1932. *Théorie des Opérations Linéaires*. Warsaw, Poland: Wydawnictwo Naukowe PWN.

Barbanel, J. 1996a. Game-theoretic algorithms for fair and strongly fair cake division with entitlements. In *Colloquium Mathematicae*, Volume 69, pp. 59–73.

Barbanel, J. 1996b. Super envy-free cake division and independence of measures. *Journal of Mathematical Analysis and Applications 197*(1): 54–60 .

Bei, X., N. Chen, X. Hua, B. Tao, and E. Yang 2012. Optimal proportional cake cutting with connected pieces. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, pp. 1263–1269. AAAI Press.

Brams, S., M. Feldman, J. Lai, J. Morgenstern, and A. Procaccia 2012. On maxsum fair cake divisions. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, pp. 1285–1291. AAAI Press.

Brams, S., M. Jones, and C. Klamler 2003. Perfect cake-cutting procedures with money. Technical report, mimeo.

Brams, S., M. Jones, and C. Klamler. 2006. Better ways to cut a cake. *Notices of the AMS 53*(11): 1314–1321 .

Brams, S., M. Jones, and C. Klamler. 2008. Proportional pie-cutting. *International Journal of Game Theory 36*(3-4): 353–367 .

Brams, S., M. Jones, and C. Klamler. 2013. N-person cake-cutting: There may be no perfect division. *The American Mathematical Monthly 120*(1): 35–47 .

Brams, S. and A. Taylor. 1995a. An envy-free cake division protocol. *The American Mathematical Monthly 102*(1): 9–18 .

Brams, S. and A. Taylor. 1995b. An envy-free cake division protocol. *The American Mathematical Monthly 102*(1): 9–18 .

Brams, S. and A. Taylor. 1996. *Fair Division: From Cake-Cutting to Dispute Resolution*. Cambridge University Press.

Brams, S., A. Taylor, and W. Zwicker. 1997. A moving-knife solution to the four-person envy-free cake-division problem. *Proceedings of the American*

*Mathematical Society 125*(2): 547–554 .

Brânzei, S., I. Caragiannis, D. Kurokawa, and A. Procaccia 2016, July. An algorithmic framework for strategic fair division. Technical Report arXiv: 1307.2225v2 [cs.GT], ACM Computing Research Repository (CoRR).

Brânzei, S. and P. Miltersen 2013. Equilibrium analysis in cake cutting. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems*, pp. 327–334. International Foundation for Autonomous Agents and Multiagent Systems.

Brânzei, S., A. Procaccia, and J. Zhang 2013. Externalities in cake cutting. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, pp. 55–61. AAAI Press.

Caragiannis, I., J. Lai, and A. Procaccia 2011. Towards more expressive cake cutting. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*, Volume 22, pp. 127.

Cechlárová, K., J. Doboš, and E. Pillárová. 2013. On the existence of equitable cake divisions. *Information Sciences* 228: 239–245 .

Cechlárová, K. and E. Pillárová. 2012a. A near equitable 2-person cake cutting algorithm. *Optimization 61*(11): 1321–1330 .

Cechlárová, K. and E. Pillárová. 2012b. On the computability of equitable divisions. *Discrete Optimization 9*(4): 249–257 .

Chen, Y., J. Lai, D. Parkes, and A. Procaccia. 2013. Truth, justice, and cake cutting. *Games and Economic Behavior 77*(1): 284–297 .

Ciesielski, K. 1989. How good is Lebesgue measure? *The Mathematical Intelligencer 11*(2): 54–58 .

Cohler, Y., J. Lai, D. Parkes, and A. Procaccia 2011. Optimal envy-free cake cutting. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence*, pp. 626–631. AAAI Press.

Deng, X., Q. Qi, and A. Saberi 2009, July. On the complexity of envy-free cake cutting. Technical Report arXiv:0907.1334v1 [cs.GT], ACM Computing Research Repository (CoRR).

Dubins, L. and E. Spanier. 1961. How to cut a cake fairly. *American Mathematical Monthly 68*(1): 1–17 .

Edmonds, J. and K. Pruhs 2006. Cake cutting really is not a piece of cake. In *Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms*, pp. 271–278.

Lebesgue, H. 1904. *Leçons sur l'Intégration.* Paris, France: Gauthier–Villars.

Lindner, C. and J. Rothe. 2009. Degrees of guaranteed envy-freeness in finite bounded cake-cutting protocols, *Proceedings of the 5th International Workshop on Internet and Network Economics*, 149–159. Springer.

Lindner, C. and J. Rothe. 2015. Cake-cutting: Fair division of divisible goods, In *Economics and Computation. An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division*, ed. Rothe, J., Springer Texts in Business and Economics, Chapter 7, 395–491. Springer-Verlag.

Maccheroni, F. and M. Marinacci. 2003. How to cut a pizza fairly: Fair division with decreasing marginal evaluations. *Social Choice and Welfare 20*(3): 457–465 .

Manabe, Y. and T. Okamoto 2010. Meta-envy-free cake-cutting protocols. In *International Symposium on Mathematical Foundations of Computer Science*, pp. 501–512. Springer.

Oxtoby, J. 1980. *Measure and Category* (2nd ed.). Graduate Texts in Mathematics. New York, USA: Springer.

Procaccia, A. 2009. Thou shalt covet thy neighbor's cake. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence*, pp. 239–244.

Procaccia, A. 2016. Cake cutting algorithms, In *Handbook of Computational Social Choice*, eds. Brandt, F., V. Conitzer, U. Endriss, J. Lang, and A. Procaccia, Chapter 13, 311–329. Cambridge University Press.

Reijnierse, J. and J. Potters. 1998. On finding an envy-free Pareto-optimal division. *Mathematical Programming 83*(1-3): 291–311 .

Robertson, J. and W. Webb. 1997. Near exact and envy free cake division. *Ars Combinatoria* 45: 97–108 .

Robertson, J. and W. Webb. 1998. *Cake-Cutting Algorithms: Be Fair If You Can.* CRC Press.

Rudin, W. 1991. *Functional Analysis* (2nd ed.). McGraw Hill.

Saberi, A. and Y. Wang. 2009. Cutting a cake for five people, *Proceedings of the 5th International Conference on Algorithmic Applications in Management*, 292–300. Springer.

Schilling, R. 2017. *Measures, Integrals and Martingales* (2nd ed.). Cambridge, UK: Cambridge University Press.

Schilling, R. and F. Kühn. 2021. *Counterexamples in Measure and Integration.* Cambridge, UK: Cambridge University Press.

Schilling, R. and D. Stoyan 2016, November. Continuity assumptions in cake-cutting. Technical Report arXiv:1611.04988v1 [cs.DS], ACM Computing Research Repository (CoRR).

Segal-Halevi, E., S. Nitzan, A. Hassidim, and Y. Aumann. 2017. Fair and square: Cake-cutting in two dimensions. *Journal of Mathematical Economics* 70: 1–28 .

Sgall, J. and G. Woeginger. 2007. An approximation scheme for cake division with a linear number of cuts. *Combinatorica 27*(2): 205–211 .

Steinhaus, H. 1948. The problem of fair division. *Econometrica* 16: 101–104 .

Stromquist, W. 2008. Envy-free cake divisions cannot be found by finite protocols. *The Electronic Journal of Combinatorics 15*(1): R11 .

Stromquist, W. and D. Woodall. 1985. Sets on which several measures agree. *Journal of Mathematical Analysis and Applications 108*(1): 241–248 .

Vitali, G. 1905. *Sul problema della misura dei Gruppi di punti di una retta: Nota.* Tip. Gamberini e Parmeggiani. Reprinted in: L. Pepe, *Giuseppe Vitali e l'analisi reale*, Rendiconti del Seminario Matematico e Fisico di Milano 54: 187–201, 1985.

von Neumann, J. 1929. Über die analytischen Eigenschaften von Gruppen linearer Transformationen und ihrer Darstellungen. *Mathematische Zeitschrift* 30: 3–42 .

Wagon, S. 1985. *The Banach–Tarski Paradox.* Cambridge, UK: Cambridge University Press.

Walsh, T. 2011. Online cake cutting, *Proceedings of the 2nd International Conference on Algorithmic Decision Theory*, 292–305. Springer.

Webb, W. 1997. How to cut a cake fairly using a minimal number of cuts. *Discrete Applied Mathematics 74*(2): 183–190 .

Woeginger, G. and J. Sgall. 2007. On the complexity of cake cutting. *Discrete Optimization 4*(2): 213–220 .

Zeng, D. 2000. Approximate envy-free procedures, *Game Practice: Contributions from Applied Game Theory*, 259–271. Springer.

# 6 Stability of Special Graph Classes

## 6.1 Summary

In this work we studied the computational complexity of decision problems related to stability of graphs. Besides some results by Papadimitriou and Wolfe [120], Cai and Meyer [35], and Burjons, Frei, Hemaspaandra, Komm, and Wehner [33], computational complexity related questions regarding the stability of graphs have not been studied intensively. In our work we built on the work by Frei, Hemaspaandra, and Rothe [67, 66], who initiated the computational complexity-theoretic investigation of this topic in 2020. Given some graph parameter $\xi$, the authors introduced, among others, the subsequent decision problems which we also studied in our work:

1. $\xi$-Stability: Given a graph $G$, the question is whether $G$ is $\xi$-stable.

2. $\xi$-VertexStability: Given a graph $G$, the question is whether $G$ is $\xi$-vertex-stable.

3. $\xi$-Unfrozenness: Given a graph $G$, the question is whether $G$ is $\xi$-unfrozen.

Although the concept of stability of graphs has a wide range of real world applications, almost all results in the aforementioned work by Frei, Hemaspaandra, and Rothe indicate that the corresponding decision problems are by no means tractably solvable. In particular, the authors proved that most of the problems are $\Theta_2^{\mathrm{p}}$-complete for the most common graph parameters.

Within our work we tried to close the gap between the desired, efficient real world usage of the concepts of stability of graphs and its high computational complexity boundaries. To do so, we studied the computational complexity of these problems from a more limited point of view. Instead of determining the computational complexity of the previously listed decision problems when allowing every possible graph as input instance, we investigated the computational complexity of these problems when restricting the set of graphs allowed as input to a single, special graph class. Particularly, we studied the complexity of these decision problems limited to trees, forests, bipartite graphs, and co-graphs as allowed input graph classes. Furthermore, we covered the same graph parameters as Frei, Hemaspaandra, and Rothe, namely the independence number $\alpha$, the vertex cover number $\beta$, the clique number $\omega$, and the chromatic number $\chi$.

With this approach we were able to formulate tractable algorithms for all previously mentioned graph parameters and every listed special graph class. We showed that these stability-related decision problems can be solved efficiently for a wide variety of graphs stemming from the investigated, special graph classes and enabled the usage of the concept of stability of graphs in real world applications that are solely based on graphs from one of these classes.

## 6.2 Publication

R. Weishaupt and J. Rothe. "Stability of Special Graph Classes". In: *Proceedings of the 22nd Italian Conference on Theoretical Computer Science*. Vol. 3072. CEUR-WS.org, Oct. 2021, pp. 234–248

A preliminary version of this work with additional results for the basic graph classes of empty graphs, complete graphs, and paths was published on *arXiv*:

R. Weishaupt and J. Rothe. *Stability of Special Graph Classes*. Tech. rep. arXiv: 2106.01496v1 [cs.CC]. June 2021

## 6.3 Personal Contribution

The writing of this work was conducted together with my co-author Jörg Rothe. All initial, technical results of this paper are my contributions.

# Stability of Special Graph Classes*

Robin Weishaupt and Jörg Rothe

Institut für Informatik, Heinrich-Heine-Universität Düsseldorf, Düsseldorf, Germany
robin.weishaupt@hhu.de, rothe@hhu.de

**Abstract.** Frei et al. [6] showed that the problem to decide whether a graph is stable with respect to some graph parameter under adding or removing either edges or vertices is $\Theta_2^{\mathrm{P}}$-complete. They studied the common graph parameters $\alpha$ (independence number), $\beta$ (vertex cover number), $\omega$ (clique number), and $\chi$ (chromatic number) for certain variants of the stability problem. We follow their approach and provide a large number of polynomial-time algorithms solving these problems for special graph classes, namely for trees, forests, bipartite graphs, and cographs.

**Keywords:** Computational Complexity · Graph Theory · Stability · Robustness · Colorability · Vertex Cover · Independent Set

## 1 Introduction

Frei et al. [6] comprehensively studied the problem of how stable certain central graph parameters are when a given graph is slightly modified, i.e., under operations such as adding or deleting either edges or vertices. Given a graph parameter $\xi$ (like, e.g., the independence number or the chromatic number), they formally introduced the problems $\xi$-Stability, $\xi$-VertexStability, $\xi$-Unfrozenness, and $\xi$-VertexUnfrozenness and showed that they are, typically, $\Theta_2^{\mathrm{P}}$-complete, that is, they are complete for the complexity class known as "parallel access to NP," which was introduced by Papadimitriou and Zachos [18] and intensely studied by, e.g., Wagner [21, 22], Hemaspaandra et al. [8, 10], and Rothe et al. [20]; see the survey by Hemaspaandra et al. [9]. $\Theta_2^{\mathrm{P}}$ is contained in the second level of the polynomial hierarchy and contains the problems that can be solved in polynomial time by an algorithm that accesses its NP oracle in parallel (i.e., it first computes all its queries and then asks them all at once and accepts its input depending on the answer vector). Alternatively, $\Theta_2^{\mathrm{P}} = \mathrm{P}^{\mathrm{NP}[\mathcal{O}(\log n)]}$ can be viewed as the class of problems solvable in polynomial time via *adaptively* accessing its NP oracle (i.e., computing the next query depending on the answer to the previous query) logarithmically often (in the input size).

Furthermore, Frei et al. [6] proved that some more specific versions of these problems, namely $k$-$\chi$-Stability and $k$-$\chi$-VertexStability, are NP-complete

for $k = 3$ and DP-complete for $k \geq 4$, respectively, where DP was introduced by Papadimitriou and Yannakakis [17] as the class of problems that can be written as the difference of NP problems.

Overall, the results of Frei et al. [6] indicate that these problems are rather intractable and there exist no efficient algorithms solving them exactly. Considering the vast number of real-world applications for these problems mentioned by Frei et al. [6]—e.g., the design of infrastructure, coloring algorithms for biological networks [15, 13] or for complex information, social, and economic networks [12], etc.—these results are rather disappointing and unsatisfying.

This obstacle motivates us to study whether there are scenarios that allow for efficient solutions to these problems which in general are intractable. Our work is based on the assumption that most of the real-world applications of stability of graph parameters do not use arbitrarily complex graphs but may often be restricted to certain special graph classes. Consequently, our studies show that—despite the completeness results by Frei et al. [6]—there are tractable solutions to these problems when one limits the scope of the problem to a special graph class. We study four different classes of special graphs: trees ($\mathcal{T}$), forests ($\mathcal{F}$), bipartite graphs ($\mathcal{B}$), and co-graphs ($\mathcal{C}$). For each such class, we study twelve different problems:

- stability, vertex-stability, and unfrozenness
- for the four graph parameters $\alpha$, $\beta$, $\omega$, and $\chi$.

In total, we thus obtain the 48 P membership results shown in Table 1, which gives the theorem, proposition, or corollary corresponding to each such P result. These results can be useful for real-world applications when knowledge about the stability, vertex-stability, or unfrozenness of a graph with respect to a certain graph parameter is required and graphs with such a special structure may typically occur in this application.

## 2 Preliminaries

We follow the notation of Frei et al. [6] and briefly collect the relevant notions here (referring to their paper [6] for further discussion). Let $\mathcal{G}$ be the set of all undirected, simple graphs without loops. For $G \in \mathcal{G}$, we denote by $V(G)$ its vertex set and by $E(G)$ its edge set; by $\overline{G}$ its complementary graph with $V(\overline{G}) = V(G)$ and $E(\overline{G}) = \{\{v, w\} \mid v, w \in V(G) \wedge v \neq w \wedge \{v, w\} \notin E(G)\}$. For $v \in V(G)$, $e \in E(G)$, and $e' \in E(\overline{G})$, let $G - v$, $G - e$, and $G + e'$, respectively, denote the graphs that result from $G$ by deleting $v$, deleting $e$, and adding $e'$.

A *graph parameter* is a map $\xi : \mathcal{G} \to \mathbb{N}$. We focus on the prominent graph parameters $\alpha$ (the size of a maximum independent set), $\beta$ (the size of a minimum vertex cover), $\chi$ (the chromatic number, i.e., the minimum number of colors needed to color the vertices of a graph so that no two adjacent vertices have the same color), and $\omega$ (the size of a maximum clique).

For a graph parameter $\xi$, an edge $e \in E(G)$ is said to be $\xi$-*stable* if $\xi(G) = \xi(G - e)$, i.e., $\xi(G)$ remains unchanged after $e$ is deleted from $G$. Otherwise (i.e.,

**Table 1.** Overview of P results established for the four special graph classes studied in this paper. E stands for the edge-related problem and V for the vertex-related problem.

| | | | $\mathcal{T}$ | $\mathcal{F}$ | $\mathcal{B}$ | $\mathcal{C}$ |
|---|---|---|---|---|---|---|
| $\alpha$ | Stability | E | Cor. 3 | Thm. 7 | Cor. 4 | Cor. 10 |
| | | V | Cor. 3 | Thm. 7 | Cor. 4 | Thm. 17 |
| | Unfrozenness | | Cor. 7 | Cor. 7 | Cor. 6 | Cor. 12 |
| $\beta$ | Stability | E | Cor. 3 | Thm. 7 | Cor. 4 | Cor. 11 |
| | | V | Cor. 3 | Thm. 7 | Cor. 4 | Cor. 8 |
| | Unfrozenness | | Cor. 7 | Cor. 7 | Thm. 12 | Cor. 12 |
| $\omega$ | Stability | E | Cor. 3 | Thm. 7 | Cor. 4 | Thm. 19 |
| | | V | Cor. 3 | Thm. 7 | Cor. 4 | Cor. 9 |
| | Unfrozenness | | Prop. 1 | Thm. 8 | Cor. 5 | Cor. 13 |
| $\chi$ | Stability | E | Cor. 3 | Thm. 7 | Cor. 4 | Thm. 18 |
| | | V | Cor. 3 | Thm. 7 | Cor. 4 | Thm. 16 |
| | Unfrozenness | | Prop. 1 | Thm. 8 | Thm. 11 | Thm. 20 |

if $\xi(G)$ is changed by deleting $e$), $e$ is said to be $\xi$-*critical*. Stability and criticality are defined analogously for a vertex $v \in V(G)$ instead of an edge $e \in E(G)$.

A graph is said to be $\xi$-*stable* if all its edges are $\xi$-stable. A graph whose vertices (instead of edges) are all $\xi$-stable is said to be $\xi$-*vertex-stable*, and $\xi$-*criticality* and $\xi$-*vertex-criticality* are defined analogously. Obviously, each edge and each vertex is either stable or critical, yet a graph might be neither.

Traditionally, the analogous terms for stability or vertex-stability when an edge or a vertex is *added* rather than deleted are *unfrozenness* and *vertex-unfrozenness*: They too indicate that a graph parameter does not change by this operation. And if, however, a graph parameter *changes* when an edge or vertex is *added* (not deleted), the notions analogous to criticality and vertex-criticality are simply termed *frozenness* and *vertex-frozenness*. Again, each edge and each vertex is either unfrozen or frozen, but a graph might be neither.

For a graph parameter $\xi$, define $\xi$-Stability to be the set of $\xi$-stable graphs; and analogously so for the sets $\xi$-VertexStability, $\xi$-VertexCriticality, $\xi$-Unfrozenness, $\xi$-Frozenness, and $\xi$-VertexUnfrozenness. These are the decision problems studied by Frei et al. [6] for general graphs in terms of their computational complexity. We will study them restricted to the graph classes mentioned in the introduction, formally defined in the subsections of Section 4.

A graph class $\mathcal{J} \subseteq \mathcal{G}$ is closed for (induced) subgraphs if for every $G \in \mathcal{J}$ it holds that all (induced) subgraphs $H$ of $G$ satisfy $H \in \mathcal{J}$.

The notation of perfect graphs was originally introduced by Berge [2] in 1963: A graph $G \in \mathcal{G}$ is called *perfect* if for all induced subgraphs $H$ of $G$, we have $\chi(H) = \omega(H)$. Note that $G$ is also an induced subgraph of itself.

Let P be the class of problems solvable in (deterministic) polynomial time. For more background on computational complexity (e.g., regarding the complexity classes NP, DP, and $\Theta_2^{\mathrm{P}}$ mentioned in the introduction; note that P $\subseteq$ NP $\subseteq$

DP $\subseteq \Theta_2^{\mathrm{P}}$ by definition), we refer to the textbooks by Papadimitriou [16] and Rothe [19].

For the sake of self-containment, we here state Gallai's theorem [7], which is used to obtain several of our results.

**Theorem 1 (Gallai's theorem).** *For every graph $G \in \mathcal{G}$, it holds that*

$$|V(G)| = \alpha(G) + \beta(G).$$

## 3 General Stability and Unfrozenness Results

In this section, we provide general results that hold for specific graph classes satisfying special requirements. These results can be used to easily determine for a given graph class whether some stability or unfrozenness results are tractable.

**Theorem 2.** *Let $\mathcal{J} \subseteq \mathcal{G}$ be a graph class closed for induced subgraphs, and $\xi$ a tractable graph parameter for $\mathcal{J}$. Then $\xi$-VertexStability $\in$ P for all $G \in \mathcal{J}$.*

**Proof.** Let $G \in \mathcal{J}$ and compute $\xi(G)$. For all $v \in V(G)$, we have $G - v \in \mathcal{J}$, since $\mathcal{J}$ is closed for induced subgraphs. Hence, for all $v \in V(G)$, we can compute $\xi(G - v)$ efficiently and compare it to $\xi(G)$. If there is no vertex such that the values differ, $G$ is $\xi$-vertex-stable. This approach is computable in time polynomial in $|G|$, so that $\xi$-VertexStability $\in$ P for all $G \in \mathcal{J}$. ❑

Since every graph class that is closed for subgraphs is also closed for induced subgraphs, Corollary 1 is a simple consequence of the previous theorem.

**Corollary 1.** *Let $\mathcal{J} \subseteq \mathcal{G}$ be a graph class closed under subgraphs and $\xi$ a tractable graph parameter for $\mathcal{J}$. Then $\xi$-VertexStability $\in$ P for all $G \in \mathcal{J}$.*

The first theorem made a statement related to vertex-stability about graph classes closed for induced subgraphs. Theorem 3 is related to stability. Due to space limitations, some proofs of our results (such as the one of Theorem 3) will be tacitly omitted; they can be found in the technical report [23].

**Theorem 3.** *Let $\mathcal{J} \subseteq \mathcal{G}$ be a graph class closed under subgraphs and $\xi$ a tractable graph parameter for $\mathcal{J}$. Then $\xi$-Stability $\in$ P for all $G \in \mathcal{J}$.*

Some of the special graph classes we study in the next section are perfect, which is why we now provide some results for perfect graph classes.

**Theorem 4.** *Let $G \in \mathcal{G}$ be a perfect graph. Then it holds that $G$ is $\omega$-vertex-stable if and only if $G$ is $\chi$-vertex-stable.*

Based on this result, the next corollary follows immediately.

**Corollary 2.** *Let $\mathcal{J} \subseteq \mathcal{G}$ be a class of perfect graphs. Then, for all graphs in $\mathcal{J}$, we have $\chi$-VertexStability $= \omega$-VertexStability.*

While the above results are related to the concepts of stability and vertex-stability, the subsequent two results address the topic of unfrozenness.

**Theorem 5.** *Let $\mathcal{J} \subseteq \mathcal{G}$ be a graph class closed under complements and subgraphs. If $\alpha$ or $\beta$ is tractable for $\mathcal{J}$, then $\omega$-*Unfrozenness $\in$ P *for all $G \in \mathcal{J}$.*

Note that this theorem exploits a relation between $\alpha$- and $\beta$-Stability and $\omega$-Unfrozenness shown in [6, Proposition 5]. The next theorem follows by a similar approach, but exploits a relation between $\omega$-Stability and $\alpha$- and $\beta$-Unfrozenness.

**Theorem 6.** *Let $\mathcal{J} \subseteq \mathcal{G}$ be a graph class closed under complements and subgraphs. If $\omega$ is tractable for $\mathcal{J}$, then $\alpha$- and $\beta$-*Unfrozenness $\in$ P *for all $G \in \mathcal{J}$.*

## 4 Tractability Results for Special Graph Classes

Ahead of our results for the individual graph classes, we provide two observations which we will use multiple times in upcoming proofs (presented here or in the technical report [23]).[1]

**Observation 1.** $\chi$-VertexStability $\subseteq$ $\chi$-Stability.

**Observation 2.** *Let $G \in \mathcal{G}$ be a graph. If an edge $\{u, v\} \in E(G)$ is $\beta$-critical, then $u$ and $v$ are $\beta$-critical, too.*

With these two observations we can now inspect several graph classes. In the following subsections we study the problems $\xi$-Stability, $\xi$-VertexStability, and $\xi$-Unfrozenness with $\xi \in \{\alpha, \beta, \omega, \chi\}$, restricted to special graph classes. Frei et al. [6] showed that for $\xi \in \{\alpha, \omega, \chi\}$ we have $\xi$-VertexUnfrozenness $= \emptyset$ as well as $\beta$-VertexUnfrozenness $= \{(\emptyset, \emptyset)\}$, where $(\emptyset, \emptyset)$ is the *null graph*, i.e., the graph without vertices or edges (which we will not further consider in this paper). This is why we do not study problems related to vertex-unfrozenness, as all related questions are already answered.

We start with investigating trees and forests.

### 4.1 Trees and Forests

We say $G \in \mathcal{G}$ is a *tree* (i.e., $G \in \mathcal{T}$) if $G$ has no isolated vertices and no cycles of length greater than or equal to 3. Furthermore, $G$ is a *forest* (i.e., $G \in \mathcal{F}$) if there exist trees $G_1, \ldots, G_n \in \mathcal{T}$ such that $G = G_1 \cup \cdots \cup G_n$. For every tree $G \in \mathcal{T}$, it holds that $|E(G)| = |V(G)| - 1$ (see, e.g., Bollobás [3]). So, we have $\omega(G) = \chi(G) = 2$ if $|V(G)| > 1$, and $\omega(G) = \chi(G) = 1$ if $|V(G)| = 1$.

Also, there exists a tractable algorithm to determine $\alpha(G)$ for trees (for example, as $\mathcal{T} \subseteq \mathcal{B}$, we can simply use the algorithm for bipartite graphs from Observation 4). Thus we can compute $\beta$ for trees using Gallai's theorem [7]

[1] Note that the second observation is in line with Observation 2 of Frei et al. [6].

(stated as Theorem 1), and all four graph parameters $\alpha$, $\beta$, $\omega$, and $\chi$ are tractable for trees.

Now, let $G \in \mathcal{F}$ with $G = G_1 \cup \cdots \cup G_n$ and $G_i \in \mathcal{T}$, $1 \leq i \leq n$, be a forest. It is easy to check that $\alpha(G) = \sum_{i=1}^{n} \alpha(G_i)$, $\beta(G) = \sum_{i=1}^{n} \beta(G_i)$, $\omega(G) = \max_{1 \leq i \leq n} \omega(G_i)$, and $\chi(G) = \max_{1 \leq i \leq n} \chi(G_i)$. Furthermore, it is known that the class of forests $\mathcal{F}$ is closed under subgraphs and induced subgraphs. From these observations we have the following results.

**Theorem 7.** *Let $\xi \in \{\alpha, \beta, \omega, \chi\}$ be a graph parameter. Then the problems $\xi$-*Stability *and $\xi$-*VertexStability *are in* P *for all forests.*

With $\mathcal{T} \subseteq \mathcal{F}$ the next corollary follows immediately.

**Corollary 3.** *For all $G \in \mathcal{T}$ and $\xi \in \{\alpha, \beta, \omega, \chi\}$, the problems $\xi$-*Stability *and $\xi$-*VertexStability *belong to* P*.*

We now focus on the unfrozenness problems. All trees and forests with fewer than three vertices are trivial to handle (see the technical report [23] for details). It remains to study trees and forests with at least three vertices.

**Proposition 1.** *Every tree $G \in \mathcal{T}$ with $|V(G)| \geq 3$ is neither $\omega$- nor $\chi$-unfrozen.*

Based on this result we can deduce whether forests are $\omega$- or $\chi$-unfrozen. As forests without edges are empty graphs, we study forests with at least one edge.

Let $P_i$ denote the path with $i$ vertices.

**Theorem 8.** *If $F \in \mathcal{F}$ contains $P_2$ but no $P_3$ as induced subgraphs, $F$ is $\omega$- and $\chi$-unfrozen. If $F$ contains $P_3$ as an induced subgraph, $F$ is not $\omega$- nor $\chi$-unfrozen.*

$\alpha$- and $\beta$-Unfrozenness are covered in Corollary 7 of the next subsection.

### 4.2 Bipartite Graphs

$G = (V_1 \cup V_2, E)$ is a *bipartite graph* if $V_1 \cap V_2 = \emptyset$ and $E \subseteq \{\{u, v\} \mid u \in V_1 \wedge v \in V_2\}$. Denote the set of all bipartite graphs by $\mathcal{B}$. We begin with two simple observations.

**Observation 3.** *Let $G \in \mathcal{B}$ be a bipartite graph. Then $\chi(G) = \omega(G) = 1$ if $E(G) = \emptyset$, and $\chi(G) = \omega(G) = 2$ if $E(G) \neq \emptyset$.*

Consequently, we can efficiently calculate $\chi$ and $\omega$ for all bipartite graphs. The proof of the next observation (see [23]) provides a tractable method to calculate $\alpha$ and $\beta$ for bipartite graphs by making use of the approaches due to Hopcroft and Karp [11] and Kőnig [14].

**Observation 4.** *We can calculate $\alpha(G)$ and $\beta(G)$ efficiently for $G \in \mathcal{B}$.*

Hence, we can efficiently calculate $\xi(G)$ for every $G \in \mathcal{B}$ and $\xi \in \{\alpha, \beta, \omega, \chi\}$. Furthermore, as the class of bipartite graphs is closed under subgraphs and induced subgraphs, the following corollary follows from Theorem 2.

**Corollary 4.** *For every $\xi \in \{\alpha, \beta, \omega, \chi\}$, the problems* $\xi$-Stability *and* $\xi$-VertexStability *are in* P *for all bipartite graphs.*

Next, we discuss approaches for how to decide whether a bipartite graph is stable. If a bipartite graph $G$ has no edges, it is trivial to handle [23]. For bipartite graphs with one edge, we have the following simple result.

**Proposition 2.** *Let $G$ be a bipartite graph with $|E(G)| = 1$. Then $G$ is neither $\xi$-stable nor $\xi$-vertex-stable for $\xi \in \{\alpha, \beta, \omega, \chi\}$.*

We now provide results for bipartite graphs with more than one edge.

**Proposition 3.** *Every bipartite graph $G$ with $|E(G)| \geq 2$ is $\chi$-stable.*

**Proof.** Let $e \in E(G)$ be an arbitrary edge of $G$. Since $E(G - e) \neq \emptyset$, it holds that $\chi(G - e) = 2 = \chi(G)$ and, thus, $G$ is $\chi$-stable. ❑

Furthermore, we can characterize $\chi$-vertex-stability.

**Theorem 9.** *Let $G$ be a bipartite graph with $|E(G)| \geq 2$. $G$ is $\chi$-vertex-stable if and only if for all $v \in V(G)$ it holds that $deg(v) < |E(G)|$.*

**Proof.** Assume $G$ to be $\chi$-vertex-stable. Furthermore, as we assume that $|E(G)| \geq 2$, it holds that $\chi(G) = 2$. Then there cannot exist a vertex $v \in V(G)$ with $\deg(v) = |E(G)|$, as such a vertex would be $\chi$-critical, since $\chi(G - v) = 1$ because of $E(G - v) = \emptyset$. For the opposite direction, assume that for all vertices $v \in V(G)$ we have $\deg(v) < |E(G)|$. Hence, no matter what vertex $v \in V(G)$ we remove from $G$, it always holds that $E(G - v) \neq \emptyset$, so $\chi(G - v) = 2 = \chi(G)$ and, thus, $G$ is $\chi$-vertex-stable. ❑

The proof of the following proposition is similar to that of Proposition 3.

**Proposition 4.** *Every bipartite graph $G$ with $|E(G)| \geq 2$ is $\omega$-stable.*

**Proof.** For all $e \in E(G)$ we have $\omega(G - e) = 2 = \omega(G)$ as $E(G - e) \neq \emptyset$ holds, such that $G$ is $\omega$-stable. ❑

Lastly, we can also characterize $\omega$-vertex-stability.

**Theorem 10.** *Let $G$ be a bipartite graph with $|E(G)| \geq 2$. $G$ is $\omega$-vertex-stable if and only if for all $v \in V(G)$ it holds that $deg(v) < |E(G)|$.*

**Proof.** Assume that $G$ is $\omega$-vertex-stable. Consequently, for all $v \in V(G)$ it holds that $\omega(G - v) = 2 = \omega(G)$. If there is one $v \in V(G)$ with $\deg(v) = |E(G)|$, we have $\omega(G - v) = 1$ as $E(G - v) = \emptyset$, a contradiction to $G$'s $\omega$-vertex-stability. Contrarily, assume that for all $v \in V(G)$ it holds that $\deg(v) < |E(G)|$. Then, for all $v \in V(G)$, it follows that $E(G - v) \neq \emptyset$. Consequently, $\omega(G) = 2 = \omega(G - v)$ and, hence, $G$ is $\omega$-vertex-stable. ❑

Besides these (vertex-)stability characterizations for bipartite graphs, we now address unfrozenness for them.

**Theorem 11.** *Let $G$ be a bipartite graph. $G$ is $\chi$-unfrozen if and only if $G$ possesses no $P_3$ as an induced subgraph.*

**Proof.** We prove both directions separately. First, assume $G$ is $\chi$-unfrozen but contains $P_3$ as an induced subgraph. Write $V(P_3) = \{v_1, v_2, v_3\}$ and $E(P_3) = \{\{v_1, v_2\}, \{v_2, v_3\}\}$ for the corresponding vertices and edges. Then $e = \{v_1, v_3\} \in \overline{E}(G)$. However, adding $e$ to $G$ we obtain $\chi(G) = 2 < 3 = \chi(G+e)$, as $P_3+e$ forms a 3-clique in $G$, a contradiction to the assumption that $G$ is $\chi$-unfrozen. Next, assume that $G$ possesses no $P_3$ as an induced subgraph but is not $\chi$-unfrozen. Hence, there must exist $e = \{u, v\} \in \overline{E}(G)$ such that $\chi(G + e) = 3 > 2 = \chi(G)$. Denote the two disjoint vertex sets of $G$ by $V_1 \cup V_2 = V(G)$. Obviously, $u \in V_1$ and $v \in V_2$ cannot be true, since then $\chi(G + e) = 2$ would hold. Therefore, without loss of generality, we assume $u, v \in V_1$. Adding $e$ to $G$ must create a cycle of odd length in $G$, as cycles of even length as well as paths can be colored with two colors. Consequently, $G + e$ possesses $C_n$ with $n = 2k + 1 \geq 3$, $k \in \mathbb{N}$, as a subgraph. This implies that $G$ must possess $P_3$ as an induced subgraph, again a contradiction. ❑

Slightly modifying (the direction from right to left in) the previous proof yields Corollary 5. This time, adding $e$ to $G$ must create a 3-clique in $G$.

**Corollary 5.** *$G \in \mathcal{B}$ is $\omega$-unfrozen if and only if $G$ possesses no $P_3$ as an induced subgraph.*

Both results show that $\omega$- and $\chi$-Unfrozenness belong to P for bipartite graphs. For the proof of Theorem 12 (see [23]), which establishes the same result for $\beta$-Unfrozenness, we need the following lemma.

**Lemma 1.** *Let $G$ be a bipartite graph and $u \in V(G)$. If $\beta(G - u) = \beta(G) - 1$, then there exists some vertex cover $V' \subseteq V(G)$ with $u \in V'$ and $|V'| = \beta(G)$.*

**Theorem 12.** *For every $G \in \mathcal{B}$, the problem $\beta$-Unfrozenness belongs to* P.

The proof of Theorem 12 allows for every nonedge of a bipartite graph to decide if it is $\beta$-unfrozen, so $\beta$-Frozenness is in P for bipartite graphs. By Gallai's theorem [7], $\alpha$-Unfrozenness is in P for bipartite graphs as well.

**Corollary 6.** *For all $G \in \mathcal{B}$, $\alpha$-Unfrozenness and $\beta$-Frozenness are in* P.

Finally, the next corollary follows by $\mathcal{T} \subseteq \mathcal{F} \subseteq \mathcal{B}$.

**Corollary 7.** *The problems $\alpha$- and $\beta$-Unfrozenness as well as the problem $\beta$-Frozenness belong to* P *for all trees and forests.*

### 4.3 Co-Graphs

First of all, we recursively define co-graphs, following a slightly adjusted definition by Corneil et al. [4].

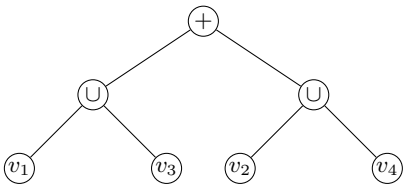



**Fig. 1.** Co-tree for $C_4$.

**Definition 1 (co-graph).** *The graph $G = (\{v\}, \emptyset)$ is a* co-graph. *If $G_1$ and $G_2$ are co-graphs, then $G_1 \cup G_2$ and $G_1 + G_2$ are co-graphs, too.*

We denote the set of all co-graphs by $\mathcal{C}$ and use the operators $\cup$ and $+$ as is common (see, e.g., [6]). We will use the following result by Corneil et al. [4].

**Theorem 13.** *Co-graphs are (i) closed under complements and (ii) closed under induced subgraphs, but (iii) not closed under subgraphs in general. Furthermore, $G \in \mathcal{G}$ is a co-graph if and only if $G$ possesses no $P_4$ as an induced subgraph.*

Property (iii) is not proven in their work [4]. However, $C_4 \in \mathcal{C}$ is an easy example since $C_4$ is a co-graph (see Example 1 below), and removing one edge yields $P_4$. Since every co-graph can be constructed by $\cup$ and $+$, we can identify a co-graph by its *co-expression*.

*Example 1 (co-expression).* The co-expression $X = (v_1 \cup v_3) + (v_2 \cup v_4)$ describes the graph $C_4 = (\{v_1, v_2, v_3, v_4\}, \{\{v_1, v_2\}, \{v_2, v_3\}, \{v_3, v_4\}, \{v_4, v_1\}\})$.

Obviously, we can build a binary tree for every co-graph via its co-expression. The tree's leaves correspond to the graph's vertices and the inner nodes of the tree correspond to the expression's operations. For example, the tree in Figure 1 corresponds to the co-expression from Example 1 and, thus, describes a $C_4$. Such a tree is called a *co-tree*. To formulate our results regarding stability and unfrozenness of co-graphs, we need the following result of Corneil et al. [5].

**Theorem 14.** *For every graph $G \in \mathcal{G}$, we can decide in $\mathcal{O}(|V(G)| + |E(G)|)$ time whether $G$ is a co-graph and, if so, provide a corresponding co-tree.*

Combining the previous results with the next one by Corneil et al. [4], we can efficiently determine a co-graph's chromatic number.

**Theorem 15.** *Let $G \in \mathcal{G}$ be a co-graph and $T$ the corresponding co-tree. For a node $w$ from $T$, denote by $G[w]$ the graph induced by the subtree of $T$ with root $w$. To every leave $v$ of $T$ we add as a label $\chi(G[v]) = 1$. For every inner node $w$ of $T$ we add, depending on the inner node's type, the following label: (1) If $w$ is a $\cup$-node with children $v_1$ and $v_2$, $\chi(G[w]) = \max\{\chi(G[v_1]), \chi(G[v_2])\}$, and (2) if $w$ is a $+$-node with children $v_1$ and $v_2$, $\chi(G[w]) = \chi(G[v_1]) + \chi(G[v_2])$. If $r$ is the root node of $T$, then it holds that $\chi(G[r]) = \chi(G)$.*

A result similar to the previous one for $\alpha$ was given by Corneil et al. [4].

*Remark 1.* We label all leaves of $T$ with $\alpha(G[v]) = 1$. Each inner node $w$ of $T$ with children $v_1$ and $v_2$ is labeled by $\alpha(G[w]) = \max\{\alpha(G[v_1]), \alpha(G[v_2])\}$ if $w$ contains the $+$-operation, and by $\alpha(G[w]) = \alpha(G[v_1]) + \alpha(G[v_2])$ if $w$ contains the $\cup$-operation. For the root $r$ of $T$, it then holds that $\alpha(G[r]) = \alpha(G)$.

By the previous remark we can efficiently calculate $\alpha$ for co-graphs. Based on these results, we can state the following theorems.

**Theorem 16.** *For every $G \in \mathcal{C}$, the problem* $\chi$-VertexStability *is in* P.

With a similar proof as for the previous theorem, we obtain the next result.

**Theorem 17.** *For every $G \in \mathcal{C}$, the problem* $\alpha$-VertexStability *is in* P.

We can use the same proof as for Theorem 17 to obtain the next corollary. However, this time we additionally use Gallai's theorem [7] to calculate $\beta$ out of $\alpha$ for $G$ and all induced subgraphs with one vertex removed.

**Corollary 8.** *For every co-graph, the problem* $\beta$-VertexStability *is in* P.

Now, the subsequent corollary follows with Proposition 5(5) by Frei et al. [6] because $\alpha$-VertexStability $= \{\overline{G} \mid G \in \omega$-VertexStability$\}$ is true and co-graphs are closed under complements.

**Corollary 9.** *For all $G \in \mathcal{C}$, the problem* $\omega$-VertexStability *is in* P.

Next, let us study the edge-related stability problems for co-graphs. To obtain our results, we need the following two auxiliary propositions.

**Proposition 5.** *Let $G \in \mathcal{C}$ with $|V(G)| > 1$ and let $u \in V(G)$ be $\chi$-critical for $G$. There exist two co-graphs $G_1, G_2$ such that $G = G_1 \cup G_2$ or $G = G_1 + G_2$. Assuming, without loss of generality, that $u \in V(G_1)$, $u$ is $\chi$-critical for $G_1$.*

**Proposition 6.** *Let $G \in \mathcal{C}$ and $e = \{u, v\} \in E(G)$. If $u$ and $v$ are $\chi$-critical for $G$, then $e$ is $\chi$-critical for $G$ as well.*

**Proof.** Let $G \in \mathcal{C}$ be a co-graph and $e = \{u, v\} \in E(G)$ an edge with two $\chi$-critical vertices $u, v \in V(G)$. First, we study the case that $G = G_1 + G_2$ as well as $u \in V(G_1)$ and $v \in V(G_2)$ holds. Afterwards, we explain how to generalize the proof.

From the previous Proposition 5 we know that $u$ must be $\chi$-critical for $G_1$ and $v$ $\chi$-critical for $G_2$. According to Observation 4 from [6] there exists an optimal coloring $c_1 \colon V(G_1) \to \mathbb{N}$ for $G_1$, such that for all $\tilde{u} \in V(G_1) \setminus \{u\}$ it holds that $c_1(\tilde{u}) \neq c_1(u)$. In other words, there is a coloring $c_1$ for $G_1$, such that $u$ is the only vertex in $G_1$ of its color. A similar, optimal coloring $c_2$ must exist for $G_2$ with respect to $v$. For the combined graph with $e$ removed, i.e., $G - e$, according to Observation 1 from [6], it must hold that $\chi(G - e) \in \{\chi(G) - 1, \chi(G)\}$. Consequently, we can reuse $c_1$ and $c_2$ from $G_1$ and $G_2$, assuming distinct colors sets for $c_1$ and $c_2$, to obtain a legal coloring of $G$ with $\chi(G)$ colors. However, we can color $u$ in the same color $c_2(v)$, as $v$ is colored, and thus obtain a legal coloring for $G - e$ with $\chi(G) - 1$ colors. This is possible because

1. $u$ is the only vertex in $G_1$ colored in $c_1(u)$ by definition of $c_1$,
2. no vertex $\tilde{u} \in V(G_1) \backslash \{u\}$ is colored with $c_2(v)$, as $c_1(V(G_1)) \cap c_2(V(G_2)) = \emptyset$ holds, and
3. $v$ is the only vertex in $G_2$ with this color, by definition of $c_2$, and there is no edge between $u$ and $v$.

Consequently, after removing $e$ from $G$, we can color $G - e$ with one color less than before, such that $\chi(G - e) = \chi(G) - 1$ holds and $e$ is $\chi$-critical.

Initially, we assumed that $G = G_1 + G_2$ with $u \in V(G_1)$ and $v \in V(G_2)$ holds. If $G = G_1 \cup G_2$, there cannot exist any edge between vertices from $G_1$ and $G_2$. Hence, the only cases left are $G = G_1 + G_2$ or $G = G_1 \cup G_2$ with both vertices in $G_1$ or $G_2$. Without loss of generality, let us assume that both vertices are in $G_1$. Following Proposition 5, we know that both vertices are $\chi$-critical for $G_1$, as they are $\chi$-critical for $G$. When we can show that $e$ is $\chi$-critical for $G_1$, it immediately follows that $e$ is also $\chi$-critical for $G$. That is because if $G = G_1 + G_2$ and $e$ is $\chi$-critical for $G_1$, we have $\chi(G_1 - e) = \chi(G_1) - 1$, such that

$$\chi(G - e) = \chi(G_1 - e) + \chi(G_2) = \chi(G_1) - 1 + \chi(G_2) = \chi(G) - 1$$

holds. If $G = G_1 \cup G_2$, there is one more argument to add. We know that $u$ and $v$ are $\chi$-critical for $G$ and $G_1$. Consequently, $\chi(G_1) > \chi(G_2)$ must hold, as otherwise, $u$ or $v$ cannot be $\chi$-critical for $G$, since $\chi(G) = \max\{\chi(G_1), \chi(G_2)\}$ is true. But then, it is enough to show that $e$ is $\chi$-critical for $G_1$, since reducing $\chi(G_1)$ by one via removing $e$ also causes a reduction of $\chi(G)$ by one and hence, $e$ is $\chi$-critical for $G$, too.

At some point, we must arrive in the case that one vertex is in $G_1$ and the other vertex is in $G_2$ and $G = G_1 + G_2$ holds, since the $+$-operation is the only possibility to add edges between vertices in co-graphs. ❑

Having these results, we are now able to provide our stability-related results.

**Theorem 18.** *For all co-graphs, the problem* $\chi$-Stability *is in* P.

**Proof.** Let $G \in \mathcal{C}$ be a co-graph. We can compute $\chi(G)$ efficiently and, according to Observation 1 in [6], for every edge $e \in E(G)$ and every vertex $v \in V(G)$ it holds that $\chi(G-e), \chi(G-v) \in \{\chi(G)-1, \chi(G)\}$. Thus, for every edge $e \in E(G)$, we proceed as follows to efficiently determine whether $e$ is $\chi$-critical or -stable for $G$: Denote $e = \{u, v\}$ for $u, v \in V(G)$. Then it follows that $G - u$ and $G - v$ are induced subgraphs of $G - e$ and $G - e$ is a subgraph of $G$. According to the earlier referenced Observation 1, it must hold that

$$\underbrace{\chi(G - v), \chi(G - u)}_{\in\{\chi(G)-1, \chi(G)\}} \leq \chi(G - e) \leq \chi(G).$$

Hence, if $\chi(G-v) = \chi(G)$ or $\chi(G-u) = \chi(G)$, which we can compute efficiently, it immediately follows that $\chi(G-e) = \chi(G)$. In other words, if $u$ or $v$ is $\chi$-stable, we know that $e$ must be $\chi$-stable, too.[2] If $u$ and $v$ are $\chi$-critical, it follows by

[2] This is in line with Observation 3 from [6].

Proposition 6 that $e$ is $\chi$-critical. Since we can determine for every node in $V(G)$ efficiently, whether it is $\chi$-stable, we can also efficiently determine for every edge in $E(G)$ whether it is $\chi$-stable. Consequently, we can decide in polynomial time whether $G$ is $\chi$-stable. Thus $\chi$-Stability $\in$ P for co-graphs follows. ❑

Next, we want to study the problem of $\omega$-Stability for co-graphs. To do so, we need the following lemmas.

**Lemma 2.** *Let $G \in \mathcal{C}$ be a co-graph. We can compute* all *cliques of size $\omega(G)$ for $G$ in time polynomial in $|G|$.*

**Lemma 3.** *Let $G \in \mathcal{G}$ be a graph and $v \in V(G)$ and $e \in E(G)$. Then it holds that $\omega(G - v)$ and $\omega(G - e)$ are in $\{\omega(G) - 1, \omega(G)\}$.*

Having these results, we can show the next theorem.

**Theorem 19.** *The problem $\omega$-Stability is in* P *for co-graphs.*

**Proof.** Let $G \in \mathcal{C}$ be a co-graph. By Theorem 15 we can compute $\omega$ efficiently for $G$ and all induced subgraphs. In order to decide whether $G$ is $\omega$-stable, we proceed as follows for every edge $e = \{u, v\} \in E(G)$:

**Case 1:** $G = G_1 \cup G_2$ for two co-graphs $G_1, G_2$, and either $u, v \in V(G_1)$ or $u, v \in V(G_2)$ holds, since there are no edges between $G_1$ and $G_2$. Assume without loss of generality that $u, v \in V(G_1)$. As $\omega(G) = \max\{\omega(G_1), \omega(G_2)\}$, we efficiently check whether $\omega(G_2) \geq \omega(G_1)$ holds. In this case, we know that $e$ cannot be critical to $G$, because even if $e$ would be $\omega$-critical to $G_1$, using Lemma 3, we still have $\omega(G - e) = \max\{\omega(G_1 - e), \omega(G_2)\} = \max\{\omega(G_1) - 1, \omega(G_2)\} = \omega(G_2)$. Otherwise, if $\omega(G_1) > \omega(G_2)$ holds, we study whether $e$ is $\omega$-critical for $G_1$ by recursively selecting the appropriate case, this time with $G_1$ as $G$. This is sufficient because if $e$ is $\omega$-critical for $G_1$, it is also $\omega$-critical for $G$.

**Case 2:** $G = G_1 + G_2$ and $u, v \in V(G_1)$ or $u, v \in V(G_2)$. In this case, it is sufficient to check whether $e$ is $\omega$-critical for the partial graph, i.e., $G_1$ or $G_2$, containing $u$ and $v$. That is because $\omega(G) = \omega(G_1) + \omega(G_2)$ holds and so, if $e$ is $\omega$-critical for one of the two partial graphs, $e$ is also critical for $G$. Once again, we check this by recursively applying the appropriate case for the corresponding partial graph.

**Case 3:** $G = G_1 + G_2$ and $u, v$ are in different partial graphs. Assume that $u \in V(G_1)$ and $v \in V(G_2)$ holds. Now, in order for $e$ to be $\omega$-critical, there must exist only one clique $V' \subseteq V(G_1)$ with $\omega(G_1) = |V'|$ as well as $u \in V'$ and only one clique $V'' \subseteq V(G_2)$ with $\omega(G_2) = |V''|$ and $v \in V''$. We can check both conditions efficiently using Lemma 2. If this is the case, then all other cliques in $G_1$ are strictly smaller than $V'$ and all other cliques in $G_2$ are strictly smaller than $V''$. Hence, the only clique of size $\omega(G)$ in $G$ is $V' \cup V''$, containing $u$ and $v$. Removing $e = \{u, v\}$ from $G$ causes $\omega(G)$ to be reduced by one since there is only one clique of size $\omega(G)$ in $G$, and afterwards, it is missing the edge $e$ in $G - e$. Therefore, only in this case $e$ is $\omega$-critical.

The number of recursive calls is limited by $\lceil \log(|V(G)|) \rceil$, since every inner node of $G$'s co-expression combines at least two nodes. Every case can be

computed efficiently, such that we can determine for a co-graph $G$ in time in $\mathcal{O}(|E(G)| \cdot \log(|V(G)|) \cdot |V(G)|^c)$ for some $c \in \mathbb{N}$ whether $G$ is $\omega$-stable. Consequently, $\omega$-Stability is in P for all co-graphs. ❑

As we now know that we can efficiently determine whether a given co-graph $G$ is $\omega$-stable, we can exploit the fact that co-graphs are closed under complements to obtain the following corollary.

**Corollary 10.** *The problem* $\alpha$-Stability *is in* P *for co-graphs.*

The next corollary follows from Gallai's theorem [7] and [6, Proposition 5].

**Corollary 11.** *The problem* $\beta$-Stability *is in* P *for co-graphs.*

At this point, we finish the study of stability problems for co-graphs, as all open questions are answered, and turn to the problems related to unfrozenness. The next two corollaries exploit the fact that co-graphs are closed under complements and follow a similar argumentation.

**Corollary 12.** *The problems* $\beta$-Unfrozenness *and* $\alpha$-Unfrozenness *are in* P *for co-graphs.*

**Corollary 13.** *The problem* $\omega$-Unfrozenness *is in* P *for co-graphs.*

Finally, we answer the last remaining open question related to unfrozenness and co-graphs.

**Theorem 20.** *The problem* $\chi$-Unfrozenness *is in* P *for co-graphs.*

**Proof.** Let $G \in \mathcal{G}$ be a co-graph and $e = \{u, v\} \in \overline{E}(G)$ a nonedge of $G$. Since $G$ has at least two vertices, $u$ and $v$, either $G = G_1 + G_2$ or $G = G_1 \cup G_2$ for two co-graphs $G_1, G_2$ holds. We handle both cases separately:

1. If $G = G_1 + G_2$ is true, then $e$ must belong either to $\overline{E}(G_1)$ or to $\overline{E}(G_2)$, since $\{\{u, v\} \mid u \in V(G_1), v \in V(G_2)\} \subseteq E(G)$, such that $\overline{E}(G) = \overline{E}(G_1) \cup \overline{E}(G_2)$. Without loss of generality assume that $e \in \overline{E}(G_1)$ holds. If $e$ is $\chi$-unfrozen for $G_1$, i.e., $\chi(G_1 + e) = \chi(G_1)$, then $e$ is $\chi$-unfrozen for $G$, since $\chi(G + e) = \chi(G_1 + e) + \chi(G_2) = \chi(G_1) + \chi(G_2) = \chi(G)$ follows. Contrarily, if $e$ is $\chi$-frozen for $G_1$, i.e., $\chi(G_1 + e) = \chi(G_1) + 1$, then $e$ is $\chi$-frozen for $G$ as well, as $\chi(G + e) = \chi(G_1 + e) + \chi(G_2) = \chi(G_1) + 1 + \chi(G_2) = \chi(G) + 1$ holds. Hence, it is enough to determine whether $e$ is $\chi$-unfrozen or -frozen for $G_1$ and we can follow the argumentation of this proof recursively for $G_1$.
2. If $G = G_1 \cup G_2$ is true, $e$ can belong to $\overline{E}(G_1), \overline{E}(G_2)$, or $\overline{E}(G) \setminus (\overline{E}(G_1) \cup \overline{E}(G_2))$. We split this into two sub-cases:
   (a) If $e \in \overline{E}(G_1)$ or $e \in \overline{E}(G_2)$, we proceed as follows. Without loss of generality assume $e \in \overline{E}(G_1)$. Since $\chi(G) = \max\{\chi(G_1), \chi(G_2)\}$ holds, an increase of $\chi(G_1)$ affects $\chi(G)$ only if $\chi(G_1) \geq \chi(G_2)$. Otherwise, $e$ is $\chi$-unfrozen for $G$ (but not necessarily for $G_1$). If $\chi(G_1) \geq \chi(G_2)$, then it holds that if $e$ is $\chi$-unfrozen for $G_1$, it follows that $e$ is $\chi$-unfrozen for $G$,

since $\chi(G+e) = \max\{\chi(G_1+e), \chi(G_2)\} = \chi(G_1+e) = \chi(G_1) = \chi(G)$ is true. Similarly, if $e$ is $\chi$-frozen for $G_1$, it follows that $e$ is $\chi$-frozen for $G$, since $\chi(G+e) = \max\{\chi(G_1+e), \chi(G_2)\} = \chi(G_1+e) = \chi(G_1)+1 = \chi(G)+1$. Consequently, it is enough to determine whether $e$ is $\chi$-unfrozen or -frozen for $G_1$ and we can follow the argumentation of this proof recursively for $G_1$.

(b) If $e \in \overline{E}(G) \setminus (\overline{E}(G_1) \cup \overline{E}(G_2))$, then $u \in V(G_1)$ and $v \in V(G_2)$ follows. Now, if $\chi(G_1) = \chi(G_2) = 1$ is true, it follows that $e$ is $\chi$-frozen for $G$, since $\chi(G+e) = 1+1 = 2 > 1 = \max\{\chi(G_1), \chi(G_2)\} = \chi(G)$. Contrarily, if $\chi(G_1) > 1$ or $\chi(G_2) > 1$, it follows that $e$ is $\chi$-unfrozen for $G$ since $G_1$ and $G_2$ share no edge but $e$. Because of that we can arrange the colors for $V(G_1)$ and $V(G_2)$ in such a way that both vertices incident to $e$ have different colors, resulting in $\chi(G+e) = \chi(G)$.

Following these cases, we can efficiently determine for every nonedge $e \in \overline{E}(G)$ whether it is $\chi$-frozen or -unfrozen for $G$, resulting in $\chi$-Unfrozenness $\in$ P for all co-graphs. ❑

## 5 Conclusion

We have provided 48 tractability results regarding the stability, vertex-stability, and unfrozenness problems when restricted to special graph classes. In particular, we studied these three problems for four important graph classes and four central graph parameters. Doing so, our work provides some baseline for further, more expanding work along this line of research. For future work, we propose to study further special graph classes that are not covered here. Besides the study of stability for other graph classes, one can also study the concept of *cost of stability*:[3] Given a graph, the question is how costly it is to stabilize it. In other words, what is the smallest number of vertices or edges to be added to or removed from the graph such that the resulting graph is stable or unfrozen with respect to some graph parameter. Relatedly, it would make sense to combine these two approaches and study the cost of stability for special graph classes.

## Acknowledgments

We thank the anonymous reviewers for helpful comments. This work was supported in part by Deutsche Forschungsgemeinschaft under grant RO 1202/21-1.

## References

1. Bachrach, Y., Elkind, E., Malizia, E., Meir, R., Pasechnik, D., Rosenschein, J., Rothe, J., Zuckerman, M.: Bounds on the cost of stabilizing a cooperative game. Journal of Artificial Intelligence Research **63**, 987–1023 (2018)

[3] Bachrach et al. [1] study a related notion of "cost of stability" for cooperative games.

2. Berge, C.: Perfect graphs. In: Six Papers on Graph Theory. pp. 1–21. Indian Statistical Institute (1963)
3. Bollobás, B.: Modern Graph Theory. Springer-Verlag (1998)
4. Corneil, D., Lerchs, H., Burlingham, L.S.: Complement reducible graphs. Discret. Appl. Math. **3**(3), 163–174 (1981)
5. Corneil, D., Perl, Y., Stewart, L.: A linear recognition algorithm for cographs. SIAM Journal on Computing **14**(4), 926–934 (1985)
6. Frei, F., Hemaspaandra, E., Rothe, J.: Complexity of stability. Journal of Computer and System Sciences (to appear), conference version appeared in the proceedings of the *31st International Symposium on Algorithms and Computation* (ISAAC'20), Leibniz International Proceedings in Informatics (LIPIcs), vol. 181, 19:1–19:14, 2020
7. Gallai, T.: Über extreme Punkt- und Kantenmengen. Ann. Univ. Sci. Budapest, Eotvos Sect. Math **2**, 133–138 (1953)
8. Hemaspaandra, E., Hemaspaandra, L., Rothe, J.: 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)
9. Hemaspaandra, E., Hemaspaandra, L., Rothe, J.: Raising NP lower bounds to parallel NP lower bounds. SIGACT News **28**(2), 2–13 (1997)
10. Hemaspaandra, E., Spakowski, H., Vogel, J.: The complexity of Kemeny elections. Theoretical Computer Science **349**(3), 382–391 (2005)
11. Hopcroft, J., Karp, R.: An $n^{5/2}$ algorithm for maximum matching in bipartite graphs. SIAM Journal on Computing **2**, 225–231 (1973)
12. Jackson, M.: Social and Economic Networks. Princeton University Press (2008)
13. Khor, S.: Application of graph coloring to biological networks. IET Systems Biology **4**(3), 185–192 (2010)
14. Kőnig, D.: Gráfok és mátrixok. Matematikai és Fizikai Lapok **38**, 116–119 (1931)
15. Minor, E., Urban, D.: A graph-theory framework for evaluating landscape connectivity and conservation planning. Conservation biology **22**(2), 297–307 (2008)
16. Papadimitriou, C.: Computational Complexity. Addison-Wesley, second edn. (1995)
17. Papadimitriou, C., Yannakakis, M.: The complexity of facets (and some facets of complexity). Journal of Computer and System Sciences **28**(2), 244–259 (1984)
18. Papadimitriou, C., Zachos, S.: Two remarks on the power of counting. In: Proceedings of the 6th GI Conference on Theoretical Computer Science. pp. 269–276. Springer-Verlag *Lecture Notes in Computer Science #145* (1983)
19. Rothe, J.: Complexity Theory and Cryptology. An Introduction to Cryptocomplexity. EATCS Texts in Theoretical Computer Science, Springer-Verlag (2005)
20. Rothe, J., Spakowski, H., Vogel, J.: Exact complexity of the winner problem for Young elections. Theory of Computing Systems **36**(4), 375–386 (2003)
21. Wagner, K.: More complicated questions about maxima and minima, and some closures of NP. Theoretical Computer Science **51**(1–2), 53–80 (1987)
22. Wagner, K.: Bounded query classes. SIAM Journal on Computing **19**(5), 833–846 (1990)
23. Weishaupt, R., Rothe, J.: Stability of special graph classes. Tech. Rep. arXiv:2106.01496 [cs.CC], arXiv.org (Jun 2021)

# 7 Conclusion

With this last chapter of our work we provide a brief overview of the results we gathered as well as an outlook for potential future work. At a glance, we studied uncertainty-related problems across four different areas of theoretical computer science, namely judgment aggregation, preference aggregation by voting, fair division of divisible goods, and stability of graphs.

In the area of judgment aggregation, we studied the family of sequential judgment aggregation rules. Having defined this family, we initiated the computational complexity study of these rules. Thereby, we proved that the winner problem belongs to $\Delta_2^{\mathrm{p}}$ for every efficiently computable, underlying base rule and is $\Delta_2^{\mathrm{p}}$-complete when using a quota rule as underlying base rule. Beyond these results, we studied the topic of manipulability via the processing order for these rules. To do so, we introduced the $S\mathcal{K}$-Winner-Design problem, the $S\mathcal{K}$-Winner-Robustness problem, as well as the $S\mathcal{K}$-Supported-Judgment problem. Our computational complexity results for these problems range from P-membership up to completeness in the second level of the polynomial-time hierarchy. Some of the previous results required a novel counting technique that we introduced and which extends the original technique by Cook [42]. Finally, we linked the studied sequential judgment aggregation rules to other, known judgment aggregation rules, such as the maximal subagenda rule [104]. In doing so, we were able to transfer some of our computational complexity related proofs and provide even stricter results for these known rules. Unfortunately, the previously mentioned complexities for the winner problem are very high, aggravating practical application of the studied rules for large instances. Hence, we suggest to follow the path by de Haan [80] and search for more tractable representations of judgment aggregation that allow for more efficient solutions to problems like the winner problem. Besides this approach, we want to intensify research on further underlying rules for sequential judgment aggregation rules. Covering a larger variety of these underlying rules improves the general understanding of the family of sequential judgment aggregation rules and might allow for more tractable approaches in the future.

With regards to preference aggregation by voting, we introduced a new decision problem $\mathcal{E}$-Possible-Winner-with-Uncertain-Weights-$\mathbb{F}$. Our problem represents a variant of the classical possible winner problem but addresses weighted elections with unspecified weights instead of unweighted elections with partial preference orders that need to be completed. Besides the initial problem, we defined three further variants thereof. The first variant contains an additional set of regions from which weights for the unweighted votes must be chosen. The second variant contains an additional upper bound that limits the overall sum of all weight to be distributed among the unweighted votes. The last variant combines the two previous variants and contains both restrictions at once. We introduced a framework to study the computational complexity of these problems for nonnegative integer as well as rational weights. With this framework at hand, we studied several established voting rules such as $k$-approval and $k$-veto for all $k \in \mathbb{N}$, plurality with

runoff, veto with runoff, Borda, simplified Bucklin and fallback voting, Copeland$^{\alpha}$, as well as ranked pairs. Making use of the technique of linear programming, we could prove P-membership for all rules and rational weights. Contrarily, our results range from P-membership up to NP-completeness for integer weights. To establish our results for plurality and veto with runoff, we proved two auxiliary results, namely that the constructive control by adding voters problem in succinct representation can be solved efficiently for both rules, i.e., it belongs to P. So far, we only covered $k$-approval, $k$-veto, and a generalization which we termed binary scoring protocol in terms of scoring protocols and nonnegative integer weights. For future work it would be desirable to establish a dichotomy result for all scoring protocols as it has been done for the original possible winner problem by Betzler and Dorn [24] and Baumeister and Rothe [23]. Beyond that, we aim to extend our study to the necessary winner with uncertain weights problem. This problem's definition would follow from the necessary winner problem in the same way as our extension follows from the related possible winner problem.

In the area of cake-cutting, we deviated from our previous approach of computational complexity studies and studied some of the very fundamental axioms of the field. In particular, we were interested in the set of all admissible pieces of cake that contains all pieces of cake which every player must be able to evaluate. We surveyed old and recent cake-cutting literature and collected several different definitions for this set. We discussed containment relations of the different approaches and provided counter examples why some of these approaches are not feasible. Thereby, our counter examples are mathematically involved and make use of abstract concepts like Banach limits and Vitali sets. Finally, we identified the set of all admissible pieces of cake with a well-known structure from measure theory, an algebra. Furthermore, based on this finding, we could identify the players' valuation functions as finite contents on such an algebra. Having these two fundamental concepts from cake-cutting identified with such thoroughly studied notions from measure theory, we argued about the optimal approach for the set of all admissible pieces of cake. We suggested to use the Borel $\sigma$-algebra over cake $X$ as set of all admissible pieces of cake. Based on this suggestion we showed that box-based valuation functions can be uniquely extended to measures on the Borel $\sigma$-algebra, allowing for a smooth transition from the previously identified, different approaches to our suggestion. In terms of future work we see two main branches to follow. First, it is interesting to think about implications of our results to cake-cutting algorithms that are used in practice. Possibly, one could identify positive consequences from the new definition for the set of all admissible pieces of cake that allow for more efficient runtimes. Second, we are interested in studying whether there are any impossibility or possibility results that are affected by our definition for the set of all admissible pieces of cake. Moreover, it could also be the case that the more profound axiomatic foundations we introduced enable new possibility or impossibility results.

Lastly, we turned the focus of our work to the area of stability of graphs. Building on the work by Frei, Hemaspaandra, and Rothe [67], we introduced a new approach by studying the stability of graphs when restricted to special graph classes. Thereby,

we studied four different graph classes, namely trees, forests, bipartite graphs, and co-graphs for the same four graph parameters as in the previously mentioned work. For all four graph classes and all four graph parameters we could define tractable algorithms that allow to determine efficiently for a given graph from one of these graph classes whether it is stable or not, vertex-stable or not, and unfrozen or not. Especially, since the stability of graphs has a wide area of applications, these results are motivating for future work. We would be interested in studying further graph classes as the four graph classes studied so far cover only a small portion of all simple graphs. Overall, we would try to aim for a dichotomy result explaining why the general stability-related decision problems are intractable compared to the restricted variants studied in our work. Additionally, we suggest to introduce another concept on top of stability of graphs, namely cost of stability. Cost of stability would describe the smallest number of edges or vertices to be added or removed from a given graph (the cost) in order to obtain a stabilized graph. This concept might help to enable even more applications of stability of graphs.

# Bibliography


[1] K. Appel and W. Haken. "Every Planar Map is 4-colorable, Part 1: Discharging". In: *Illinois Journal of Mathematics* 21 (1977), pp. 429–490.

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

[3] K. Arrow. *Social Choice and Individual Values*. 3rd ed. Yale University Press, 2012.

[4] K. Arrow, A. Sen, and K. Suzumura, eds. *Handbook of Social Choice and Welfare*. Vol. 2. North-Holland, 2011.

[5] H. Aziz. "Developments in Multi-Agent Fair Allocation". In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 34. 09. 2020, pp. 13563–13568.

[6] H. Aziz, M. Brill, F. Fischer, P. Harrenstein, J. Lang, and H. Seedig. "Possible and necessary winners of partial tournaments". In: *Journal of Artificial Intelligence Research* 54 (2015), pp. 493–534.

[7] H. Aziz and S. Mackenzie. "A Discrete and Bounded Envy-Free Cake Cutting Protocol for Any Number of Agents". In: *2016 IEEE 57th Annual Symposium on Foundations of Computer Science (FOCS)*. IEEE. 2016, pp. 416–427.

[8] Y. Bachrach, N. Betzler, and P. Faliszewski. "Probabilistic Possible Winner Determination". In: *Proceedings of the 24th AAAI Conference on Artificial Intelligence*. AAAI Press, July 2010, pp. 697–702.

[9] J. Barbanel. "Super Envy-Free Cake Division and Independence of Measures". In: *Journal of Mathematical Analysis and Applications* 197.1 (1996), pp. 54–60.

[10] J. Bartholdi III, C. Tovey, and M. Trick. "How Hard Is It to Control an Election?" In: *Mathematical and Computer Modelling* 16.8/9 (1992), pp. 27–40.

[11] J. Bartholdi III, C. Tovey, and M. Trick. "The Computational Difficulty of Manipulating an Election". In: *Social Choice and Welfare* 6.3 (1989), pp. 227–241.

[12] J. Bartholdi III, C. Tovey, and M. Trick. "Voting Schemes for Which it Can Be Difficult to Tell Who Won the Election". In: *Social Choice and Welfare* 6.2 (1989), pp. 157–165.

[13] D. Bauer, F. Harary, J. Nieminen, and C. Suffel. "Domination alteration sets in graphs". In: *Discrete Mathematics* 47 (1983), pp. 153–161.

[14] D. Baumeister, L. Boes, and R. Weishaupt. "Complexity of Sequential Rules in Judgment Aggregation". In: *Proceedings of the 20th International Conference on Autonomous Agents and Multiagent Systems*. 2021, pp. 187–195.

[15] D. Baumeister, L. Boes, and R. Weishaupt. "Complexity of Sequential Rules in Judgment Aggregation". In: *The 8th International Workshop on Computational Social Choice (COMSOC-21)*. Ed. by B. Zwicker and R. Meir. Available online at `https://comsoc2021.net.technion.ac.il/accepted-papers/`. Haifa, Israel: Technion-Israel Institute of Technology, 2021.

[16] D. Baumeister, G. Erdélyi, O. Erdélyi, and J. Rothe. "Complexity of manipulation and bribery in judgment aggregation for uniform premise-based quota rules". In: *Mathematical Social Sciences* 76 (2015), pp. 19–30.

[17] D. Baumeister, G. Erdélyi, and J. Rothe. "Judgment Aggregation". In: *Economics and Computation. An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division*. Ed. by J. Rothe. Springer Texts in Business and Economics. Springer-Verlag, 2015. Chap. 6, pp. 361–391.

[18] D. Baumeister, P. Faliszewski, J. Lang, and J. Rothe. "Campaigns for Lazy Voters: Truncated Ballots". In: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems*. IFAAMAS, 2012, pp. 577–584.

[19] D. Baumeister, M. Neveling, M. Roos, J. Rothe, L. Schend, R. Weishaupt, and L. Xia. "The Possible Winner with Uncertain Weights Problem". In: *Journal of Computer and System Sciences* (Submitted).

[20] 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*. IFAAMAS, 2011, pp. 853–860.

[21] 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*. IOS Press, Aug. 2012, pp. 133–138.

[22] D. Baumeister and J. Rothe. "Preference Aggregation by Voting". In: *Economics and Computation. An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division*. Ed. by J. Rothe. Springer Texts in Business and Economics. Springer-Verlag, 2015. Chap. 4, pp. 197–325.

[23] D. Baumeister and J. Rothe. "Taking the final step to a full dichotomy of the possible winner problem in pure scoring rules". In: *Information Processing Letters* 112.5 (2012), pp. 186–190.

[24] N. Betzler and B. Dorn. "Towards a dichotomy for the Possible Winner problem in elections based on scoring rules". In: *Journal of Computer and System Sciences* 76.8 (2010), pp. 812–836.

[25] N. Betzler, R. Niedermeier, and G. Woeginger. "Unweighted Coalitional Manipulation Under the Borda Rule Is NP-Hard". In: *Proceedings of the 22nd International Joint Conference on Artificial Intelligence*. AAAI Press/IJCAI, July 2011, pp. 55–60.

[26] H. Bodlaender and A. Koster. "Combinatorial optimization on graphs of bounded treewidth". In: *The Computer Journal* 51.3 (2008), pp. 255–269.

[27] B. Bollobás. *Modern Graph Theory*. Springer-Verlag, 1998.

[28] J. Borda. "Mémoire sur les élections au scrutin". In: *Histoire de L'Académie Royale des Sciences, Paris* (1781). English translation appears in the paper by de Grazia [74].

[29] S. Brams and R. Sanver. "Voting Systems that Combine Approval and Preference". In: *The Mathematics of Preference, Choice, and Order: Essays in*

*Honor of Peter C. Fishburn*. Ed. by S. Brams, W. Gehrlein, and F. Roberts. Springer, 2009, pp. 215–237.

[30] S. Brams and A. Taylor. *Fair Division: From Cake-Cutting to Dispute Resolution*. Cambridge University Press, 1996.

[31] S. Brams, A. Taylor, and W. Zwicker. "Old and New Moving-Knife Schemes". In: *The Mathematical Intelligencer* 17.4 (1995), pp. 30–35.

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

[33] E. Burjons, F. Frei, E. Hemaspaandra, D. Komm, and D. Wehner. "Finding Optimal Solutions With Neighborly Help". In: *Proceedings of the 44th International Symposium on Mathematical Foundations of Computer Science*. Vol. 138. Leibniz International Proceedings in Informatics (LIPIcs). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2019, 78:1–78:14.

[34] J. Cai. "$S_2^p \subseteq ZPP^{NP}$". In: *Journal of Computer and System Sciences* 73 (2007), pp. 25–35.

[35] J. Cai and G. Meyer. "Graph Minimal Uncolorability is $D^P$-complete". In: *SIAM Journal on Computing* 16.2 (1987), pp. 259–277.

[36] R. Canetti. "More on BPP and the Polynomial-Time Hierarchy". In: *Information Processing Letters* 57.5 (1996), pp. 237–241.

[37] F. Cariani. "Judgment aggregation". In: *Philosophy compass* 6.1 (2011), pp. 22–32.

[38] Y. Chevaleyre, P. Dunne, U. Endriss, J. Lang, M. Lemaître, N. Maudet, J. Padget, S. Phelps, J. Rodríguez-Aguilar, and P. Sousa. "Issues in Multiagent Resource Allocation". In: *Informatica* 30.1 (2006), pp. 3–31.

[39] 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*. AAAI Press, July 2010, pp. 762–767.

[40] V. Conitzer, T. Sandholm, and J. Lang. "When Are Elections with Few Candidates Hard to Manipulate?" In: *Journal of the ACM* 54.3 (2007), 14–es.

[41] V. Conitzer and T. Walsh. "Barriers to Manipulation in Voting". In: *Handbook of Computational Social Choice*. Ed. by F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. Procaccia. Cambridge University Press, 2016. Chap. 6, pp. 127–145.

[42] S. Cook. "The Complexity of Theorem-Proving Procedures". In: *Proceedings of the 3rd ACM Symposium on Theory of Computing*. ACM Press, 1971, pp. 151–158.

[43] S. Cook. "The P versus NP problem". In: *The Millennium Prize Problems*. Ed. by J. Carlson, A. Jaffe, and A. Wiles. American Mathematical Society, 2006, pp. 87–104.

[44] A. Copeland. "A "Reasonable" Social Welfare Function". Mimeographed notes from a Seminar on Applications of Mathematics to the Social Sciences, University of Michigan. 1951.

[45] D. Corneil, H. Lerchs, and L. S. Burlingham. "Complement reducible graphs". In: *Discrete Applied Mathematics* 3 (1981), pp. 163–174.

[46] J. Davies, G. Katsirelos, N. Narodytska, T. Walsh, and L. Xia. "Complexity of and algorithms for the manipulation of Borda, Nanson's and Baldwin's voting rules". In: *Artificial Intelligence* 217 (2014), pp. 20–42.

[47] R. Diestel. *Graph Decompositions: A Study in Infinite Graph Theory*. Oxford University Press, 1990.

[48] F. Dietrich and C. List. "Arrow's theorem in judgment aggregation". In: *Social Choice and Welfare* 29.1 (2007), pp. 19–33.

[49] F. Dietrich and C. List. "Judgment Aggregation by Quota Rules: Majority Voting Generalized". In: *Journal of Theoretical Politics* 19.4 (2007), pp. 391–424.

[50] F. Dietrich and C. List. "Strategy-proof Judgment Aggregation". In: *Economics and Philosophy* 23.3 (2007), pp. 269–300.

[51] G. Dirac. "Some theorems on abstract graphs". In: *Proceedings of the London Mathematical Society* 3.1 (1952), pp. 69–81.

[52] L. Dubins and E. Spanier. "How to Cut a Cake Fairly". In: *The American Mathematical Monthly* 68.1 (1961), pp. 1–17.

[53] U. Endriss. "Judgment Aggregation". In: *Handbook of Computational Social Choice*. Ed. by F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. Procaccia. Cambridge University Press, 2016. Chap. 17, pp. 399–426.

[54] U. Endriss, U. Grandi, R. de Haan, and J. Lang. "Succinctness of Languages for Judgment Aggregation". In: *Proceedings of the 15th International Conference on Principles of Knowledge Representation and Reasoning*. AAAI Press, Apr. 2016, pp. 176–185.

[55] U. Endriss, U. Grandi, and D. Porello. "Complexity of Judgment Aggregation". In: *Journal of Artificial Intelligence Research* 45 (2012), pp. 481–514.

[56] U. Endriss and R. de Haan. "Complexity of the Winner Determination Problem in Judgment Aggregation: Kemeny, Slater, Tideman, Young". In: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems*. IFAAMAS, 2015, pp. 117–125.

[57] U. Endriss, R. de Haan, J. Lang, and M. Slavkovik. "The Complexity Landscape of Outcome Determination in Judgment Aggregation". In: *Journal of Artificial Intelligence Research* 69 (2020), pp. 687–731.

[58] G. Erdélyi, M. Fellows, J. Rothe, and L. Schend. "Control Complexity in Bucklin and fallback voting: A theoretical analysis". In: *Journal of Computer and System Sciences* 81.4 (2015), pp. 632–660.

[59] G. Erdélyi, M. Fellows, J. Rothe, and L. Schend. "Control Complexity in Bucklin and fallback voting: An experimental analysis". In: *Journal of Computer and System Sciences* 81.4 (2015), pp. 661–670.

[60] G. Erdélyi, M. Neveling, C. Reger, J. Rothe, Y. Yang, and R. Zorn. "Towards completing the puzzle: complexity of control by replacing, adding, and deleting candidates or voters". In: *Autonomous Agents and Multi-Agent Systems* 35.2 (2021), pp. 1–48.

[61] L. Euler. "Solutio problematis ad geometriam situs pertinentis". In: *Commentarii academiae scientiarum Petropolitanae* (1741), pp. 128–140.

[62] S. Even and A. Paz. "A Note on Cake Cutting". In: *Discrete Applied Mathematics* 7.3 (1984), pp. 285–296.

[63] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. "The Complexity of Bribery in Elections". In: *Proceedings of the 21st National Conference on Artificial Intelligence*. AAAI Press, July 2006, pp. 641–646.

[64] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. "Using Complexity to Protect Elections". In: *Communications of the ACM* 53.11 (2010), pp. 74–82.

[65] Z. Fitzsimmons and E. Hemaspaandra. "High-multiplicity election problems". In: *Autonomous Agents and Multi-Agent Systems* 33.4 (2019), pp. 383–402.

[66] F. Frei, E. Hemaspaandra, and J. Rothe. "Complexity of Stability". In: *Proceedings of the 31st International Symposium on Algorithms and Computation*. Vol. 181. Leibniz International Proceedings in Informatics (LIPIcs). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dec. 2020, 19:1–19:14.

[67] F. Frei, E. Hemaspaandra, and J. Rothe. "Complexity of stability". In: *Journal of Computer and System Sciences* 123 (2022), pp. 103–121.

[68] T. Gallai. "Über extreme Punkt- und Kantenmengen". In: *Annales Univ. Sci. Budapest* 2 (1959), pp. 133–138.

[69] R. Gandy. "Church's Thesis and Principles for Mechanisms". In: *Studies in Logic and the Foundations of Mathematics*. Vol. 101. Elsevier, 1980, pp. 123–148.

[70] V. Ganesan, M. Slavkovik, S. Sousa, and L. van der Torre. "Judgment Aggregation for Cooperative Anchoring on the NAO Robots". In: *Works-in-progress track - 3rd International Conference on Social Robotics*. 2011.

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

[72] A. Gibbard. "Manipulation of Voting Schemes: A General Result". In: *Econometrica* 41.4 (1973), pp. 587–601.

[73] U. Grandi and U. Endriss. "Lifting integrity constraints in binary aggregation". In: *Artificial Intelligence* 199-200 (2013), pp. 45–66.

[74] A. de Grazia. "Mathematical Derivation of an Election System". In: *Isis* 44.1–2 (1953), pp. 42–51.

[75] G. Gunther, B. Hartnell, and D. Rall. "Graphs whose vertex independence number is unaffected by single edge addition or deletion". In: *Discrete Applied Mathematics* 46.2 (1993), pp. 167–172.

[76] R. Gupta. "The chromatic index and the degree of a graph". In: *Notices of the American Mathematical Society* 13.719 (1966), p. 449.

[77] F. Gurski. "Characterizations for co-graphs defined by restricted NLC-width or clique-width operations". In: *Discrete Mathematics* 306.2 (2006), pp. 271–277.

[78] F. Gurski and C. Rehs. "Directed Path-Width and Directed Tree-Width of Directed Co-graphs". In: *Computing and Combinatorics*. Ed. by L. Wang and D. Zhu. Springer, 2018, pp. 255–267.

[79] F. Gurski and E. Wanke. "On the relationship between NLC-width and linear NLC-width". In: *Theoretical Computer Science* 347.1-2 (2005), pp. 76–89.

[80] R. de Haan. "Hunting for Tractable Languages for Judgment Aggregation". In: *Proceedings of the 16th International Conference on Principles of Knowledge Representation and Reasoning*. AAAI Press, 2018, pp. 194–203.

[81] R. de Haan and M. Slavkovik. "Complexity Results for Aggregating Judgments using Scoring or Distance-Based Procedures". In: *Proceedings of the 16th Conference on Autonomous Agents and Multiagent Systems*. 2017, pp. 952–961.

[82] L. Hačijan. "A Polynomial Algorithm in Linear Programming". In: *Soviet Mathematics Doklady* 20.1 (1979), pp. 191–194.

[83] F. Harary. *Graph Theory*. Addison-Wesley, 1969.

[84] J. Hartmanis and R. Stearns. "On the Computational Complexity of Algorithms". In: *Transactions of the American Mathematical Society* 117 (1965), pp. 285–306.

[85] E. Hemaspaandra and L. Hemaspaandra. "Dichotomy for voting systems". In: *Journal of Computer and System Sciences* 73.1 (2007), pp. 73–83.

[86] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. "The Complexity of Manipulative Actions in Single-Peaked Societies". In: *Economics and Computation. An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division*. Ed. by J. Rothe. Springer Texts in Business and Economics. Springer-Verlag, 2015. Chap. 5, pp. 327–360.

[87] E. Hemaspaandra, L. Hemaspaandra, and H. Schnoor. "A Control Dichotomy for Pure Scoring Rules". In: *Proceedings of the 28th AAAI Conference on Artificial Intelligence*. AAAI Press, July 2014, pp. 712–720.

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

[89] L. Hemaspaandra and M. Ogihara. *The Complexity Theory Companion*. EATCS Texts in Theoretical Computer Science. Springer-Verlag, 2002.

[90] M. Henning and M. Krzywkowski. "Total domination stability in graphs". In: *Discrete Applied Mathematics* 236 (2018), pp. 246–255.

[91] R. Karp. "Reducibility among Combinatorial Problems". In: *Complexity of Computer Computations*. Ed. by R. Miller, J. Thatcher, and J. Bohlinger. Springer, 1972, pp. 85–103.

[92] R. Karp and R. Lipton. "Some Connections Between Nonuniform and Uniform Complexity Classes". In: *Proceedings of the 12th ACM Symposium on Theory of Computing*. ACM Press, Apr. 1980, pp. 302–309.

[93] J. Kemeny. "Mathematics Without Numbers". In: *Dædalus* 88.4 (1959), pp. 577–591.

[94] P. Kern, D. Neugebauer, J. Rothe, R. Schilling, D. Stoyan, and R. Weishaupt. "A Closer Look at the Cake-Cutting Foundations through the Lens of Measure Theory". In: *The 8th International Workshop on Computational Social Choice (COMSOC-21)*. Ed. by B. Zwicker and R. Meir. Available online at `https://comsoc2021.net.technion.ac.il/accepted-papers/`. Haifa, Israel: Technion-Israel Institute of Technology, 2021.

[95] P. Kern, D. Neugebauer, J. Rothe, R. Schilling, D. Stoyan, and R. Weishaupt. *Cutting a Cake Is Not Always a "Piece of Cake": A Closer Look at the Foundations of Cake-Cutting Through the Lens of Measure Theory*. Tech. rep. arXiv: 2111.05402v1 [cs.GT]. Nov. 2021.

[96] P. Kern, D. Neugebauer, J. Rothe, R. Schilling, D. Stoyan, and R. Weishaupt. "Cutting a Cake Is Not Always a "Piece of Cake": A Closer Look at the Foundations of Cake-Cutting Through the Lens of Measure Theory". In: *Social Choice and Welfare* (Submitted).

[97] D. Knuth. "Big omicron and big omega and big theta". In: *ACM SIGACT News* 8.2 (1976), pp. 18–24.

[98] K. Konczak and J. Lang. "Voting Procedures with incomplete preferences". In: *Proceedings of the Multidisciplinary IJCAI-05 Workshop on Advances in Preference Handling*. 2005, pp. 124–129.

[99] L. Kornhauser and L. Sager. "Unpacking the Court". In: *Yale Law Journal* 96.1 (1986), pp. 82–117.

[100] M. Krentel. "The Complexity of Optimization Problems". In: *Journal of Computer and System Sciences* 36 (1988), pp. 490–509.

[101] H. Kuhn. "On Games of Fair Division". In: *Essays in Mathematical Economics, in Honor of Oskar Morgenstern*. Ed. by M. Shubik. Princeton University Press, 1967, pp. 29–37.

[102] J. Lang. "Collective Decision Making under Incomplete Knowledge: Possible and Necessary Solutions". In: *Proceedings of the 29th International Joint Conference on Artificial Intelligence*. ijcai.org, July 2020, pp. 4885–4891.

[103] J. Lang and J. Rothe. "Fair Division of Indivisible Goods". In: *Economics and Computation. An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division*. Ed. by J. Rothe. Springer Texts in Business and Economics. Springer-Verlag, 2015. Chap. 8, pp. 493–550.

[104] J. Lang and M. Slavkovik. "How Hard is it to Compute Majority-Preserving Judgment Aggregation Rules?" In: *Proceedings of the 21st European Conference on Artificial Intelligence*. IOS Press. 2014, pp. 501–506.

[105] C. Lautemann. "BPP and the Polynomial Hierarchy". In: *Information Processing Letters* 17.4 (1983), pp. 215–217.

[106] H. Lerchs. "On cliques and kernels". In: *Department of Computer Science, University of Toronto* 1 (1971).

[107] H. Lerchs. "On the clique-kernel structure of graphs". In: *Department of Computer Science, University of Toronto* 1 (1972).

[108] L. Levin. "Universal Sorting Problems". In: *Problemy Peredaci Informacii* 9 (1973). In Russian. English translation by B. Trakhtenbrot [135], pp. 115–116.

[109] C. Lindner and J. Rothe. "Cake-Cutting: Fair Division of Divisible Goods". In: *Economics and Computation. An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division*. Ed. by J. Rothe. Springer Texts in Business and Economics. Springer-Verlag, 2015. Chap. 7, pp. 395–491.

[110] C. List. "A Model of Path-Dependence in Decisions over Multiple Propositions". In: *American Political Science Review* 98.3 (2004), pp. 495–513.

[111] C. List. "The Discursive Dilemma and Public Reason". In: *Ethics* 116.2 (2006), pp. 362–402.

[112] C. List and P. Pettit. "Aggregating Sets of Judgments: An Impossibility Result". In: *Economics and Philosophy* 18.1 (2002), pp. 89–110.

[113] C. List and C. Puppe. "Judgment Aggregation". In: *The Handbook of Rational and Social Choice*. Ed. by P. Anand, P. Pattanaik, and C. Puppe. Oxford University Press, 2009. Chap. 19, pp. 457–482.

[114] K. Nehring and C. Puppe. "The structure of strategy-proof social choice. Part I: General characterization and possibility results on median space". In: *Journal of Economic Theory* 135.1 (2007), pp. 269–305.

[115] K. Nehring and C. Puppe. "The Structure of Strategy-Proof Social Choice. Part II: Non-Dictatorship, Anonymity, and Neutrality". Unpublished manuscript. Available online at `https://micro.econ.kit.edu/english/536.php`. Mar. 2005.

[116] M. Neveling and J. Rothe. "Control complexity in Borda elections: Solving all open cases of offline control and some cases of online control". In: *Artificial Intelligence* 298 (2021), p. 103508.

[117] M. Neveling, J. Rothe, and R. Weishaupt. "The Possible Winner Problem with Uncertain Weights Revisited". In: *Proceedings of the 23rd International Symposium on Fundamentals of Computation Theory*. Springer-Verlag LNCS, Sept. 2021, pp. 399–412.

[118] T. Nguyen, M. Roos, and J. Rothe. "A survey of approximability and inapproximability results for social welfare optimization in multiagent resource allocation". In: *Annals of Mathematics and Artificial Intelligence* 68.1–3 (2013), pp. 65–90.

[119] C. Papadimitriou. *Computational Complexity*. 2nd ed. Addison-Wesley, 1995.

[120] C. Papadimitriou and D. Wolfe. "The Complexity of Facets Resolved". In: *Journal of Computer and System Sciences* 37.1 (1988), pp. 2–13.

[121] C. Papadimitriou and S. Zachos. "Two Remarks on the Power of Counting". In: *Proceedings of the 6th GI Conference on Theoretical Computer Science*. Vol. 145. Lecture Notes in Computer Science. Springer, 1983, pp. 269–276.

[122] P. Pettit. "Deliberative Democracy and the Discursive Dilemma". In: *Philosophical Issues* 11.1 (2001), pp. 268–299.

[123] O. Pikhurko. "On Envy-Free Cake Division". In: *The American Mathematical Monthly* 107.8 (2000), pp. 736–738.

[124] J. Robertson and W. Webb. *Cake-Cutting Algorithms: Be Fair If You Can*. A K Peters, 1998.

[125] J. Rothe. "Borda Count in Collective Decision Making: A Summary of Recent Results". In: *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*. AAAI Press, 2019, pp. 9830–9836.

[126] J. Rothe. *Complexity Theory and Cryptology. An Introduction to Cryptocomplexity*. EATCS Texts in Theoretical Computer Science. Springer-Verlag, 2005.

[127] J. Rothe, ed. *Economics and Computation. An Introduction to Algorithmic Game Theory, Computational Social Choice, and Fair Division*. Springer Texts in Business and Economics. Springer-Verlag, 2015.

[128] M. Satterthwaite. "Strategy-Proofness and Arrow's Conditions: Existence and Correspondence Theorems for Voting Procedures and Social Welfare Functions". In: *Journal of Economic Theory* 10.2 (1975), pp. 187–217.

[129] M. Schaefer and C. Umans. "Completeness in the Polynomial-Time Hierarchy: Part I: A Compendium". In: *SIGACT News* 33.3 (Sept. 2002), pp. 32–49.

[130] M. Schaefer and C. Umans. "Completeness in the Polynomial-Time Hierarchy: Part II". In: *SIGACT News* 33.4 (Dec. 2002), pp. 22–36.

[131] K. Skiba, D. Neugebauer, and J. Rothe. "Complexity of Nonempty Existence Problems in Incomplete Argumentation Frameworks". In: *IEEE Intelligent Systems* 36.2 (2021), pp. 13–24.

[132] H. Steinhaus. "The Problem of Fair Division". In: *Econometrica* 16 (1948), pp. 101–104.

[133] L. Stockmeyer. "The Polynomial-Time Hierarchy". In: *Theoretical Computer Science* 3.1 (1977), pp. 1–22.

[134] C. Tovey. "Tutorial on Computational Complexity". In: *Interfaces* 32.3 (2002), pp. 30–61.

[135] B. Trakhtenbrot. "A Survey of Russian Approaches to Perebor (Brute-Force Search) Algorithms". In: *Annals of the History of Computing* 6.4 (1984), pp. 384–400.

[136] A. Turing. "On Computable Numbers, with an Application to the Entscheidungsproblem". In: *Proceedings of the London Mathematical Society* s2-42.1 (1936). Correction, *ibid*, Vol. s2-43, pp. 544–546, 1937, pp. 230–265.

[137] C. Umans. "Hardness of Approximating $\Sigma_2^p$ Minimization Problems". In: *40th Annual Symposium on Foundations of Computer Science*. IEEE. 1999, pp. 465–474.

[138] V. Vizing. "On an estimate of the chromatic class of a $p$-graph". In: *Diskret. Analiz* 3 (1964), pp. 25–30.

[139] W. Webb. "An Algorithm For Super Envy-Free Cake Division". In: *Journal of Mathematical Analysis and Applications* 239.4 (1999), pp. 175–179.

[140] R. Weishaupt and J. Rothe. *Stability of Special Graph Classes*. Tech. rep. arXiv: 2106.01496v1 [cs.CC]. June 2021.

[141] R. Weishaupt and J. Rothe. "Stability of Special Graph Classes". In: *Proceedings of the 22nd Italian Conference on Theoretical Computer Science*. Vol. 3072. CEUR-WS.org, Oct. 2021, pp. 234–248.

[142] D. West. *Introduction to Graph Theory*. Vol. 2. Prentice Hall, 2001.

[143] L. Xia and V. Conitzer. "Determining Possible and Necessary Winners under Common Voting Rules Given Partial Orders". In: *Journal of Artificial Intelligence Research* 41 (2011), pp. 25–67.

[144] S. Xu. "Relations between parameters of a graph". In: *Discrete Mathematics* 89.1 (1991), pp. 65–88.

# Eidesstattliche Versicherung

*Gemäß §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.

Darüber hinaus erkläre ich, dass ich die Dissertation in der vorliegenden oder in ähnlicher Form noch bei keiner anderen Institution eingereicht habe.

Teile dieser Dissertation wurden bereits in Form von Zeitschriftenartikeln und Konferenzberichten veröffentlicht oder zur Begutachtung eingereicht und sind entsprechend referenziert: [14], [15], [21], [117], [19], [94] [95], [96], [140] und [141].

Düsseldorf, den 17. März 2022 Robin Weishaupt