From Individual to Collective

Exploring Multiwinner Elections, Participatory Budgeting, and Judgment Aggregation on a Local and Global Level

Inaugural-Dissertation

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

vorgelegt von

Linus Leopold Boes

geboren in Solingen

Düsseldorf, März 2024

Heinrich Heine
Universität
Düsseldorf

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. Frau Prof. Dr. Dorothea Baumeister
2. Herr Prof. Dr. Jörg Rothe

Tag der mündlichen Prüfung: 19. 04. 2024

Abstract

In this thesis, we explore three structurally close frameworks for aggregating individual (approval-based) preferences into a collective outcome, namely *multiwinner elections*, *participatory budgeting*, and *judgment aggregation*. Located at the core of *computational social choice*, we study aggregation mechanisms for multiagent decision-making processes from a computational point of view, using tools and techniques from theoretical computer science and artificial intelligence. In the respective settings, a set of agents decides, which subset of predefined alternatives should appear in a collective outcome. For this purpose, each agent provides an individual binary evaluation on whether each distinct alternative should be selected. A constraint limits the set of feasible outcomes and a voting rule maps the agents' preferences to at least one feasible set of winning alternatives.

Motivated by different practical applications, the three aforementioned frameworks can be informally distinguished as follows. In multiwinner elections, voters elect a fixed-size committee of candidates. In participatory budgeting, citizens decide over the spending of limited public funds on a selection of projects, each having a predefined cost for implementation. In judgment aggregation, a set of judges must come to a (logically consistent) agreement over the truthfulness of a set of logically interconnected propositions.

Overall, the key results acquired in this thesis can be grouped into five categories: An *(i) axiomatic analysis*, helps understanding the behavior of voting rules and their limitations. Notably, we develop multiple impossibility results (stating that some properties cannot be satisfied simultaneously by any voting method) and discuss potential ways to partly escape those negative results. To select a voting rule that can realistically be used in an election (by computing an outcome in a reasonable amount of time), it is important to study the *(ii) computational complexity of winner determination* in the first place. Our results for related decision problems range from efficiently computable algorithms to hardness in the second level of the polynomial-time hierarchy. Contrarily, if a voting rule is prone to some kind of *(iii) manipulative interference*, a high complexity for computing a beneficial manipulative action might render strategic behavior infeasible. Along with identifying the complexity for various related decision problems, we discuss ways to prevent manipulative actions for efficiently computable rules. We uncover several *(iv) relationships between voting rules*, linking problems that have been studied independently closer together. Lastly, we investigate the potential impact from considering a slightly generalized *(v) ballot design*.

In loving memory of my grandparents Jutta and Hans-Gerd,
who were not able to celebrate my biggest achievements with me.

Acknowledgments

From a very early age on, I was fascinated by computers and programming. Eventually, a special issue of the (German version of the) magazine *Scientific American* sparked my interest in the beauty of theoretical computer science and mathematics. Coincidentally, this issue contained articles about cake-cutting and game theory. Years later, I learned that similar topics, revolving around decision-making processes and human behavior, are actually pursued academically at the *Heinrich-Heine-Universität*. I immediately signed up for my Master's degree, which luckily turned into a doctorate position in computational social choice. Looking back at those exciting past five years of graduation, let me express my gratitude for all the extraordinary people that helped me along the way.

First and foremost, I thank my supervisor Doro for trusting in my abilities and for leading as an exceptional role model. Thank you for providing me with the perfect ratio of challenge and support day and night. I am grateful for everything you have taught me — to ask relevant questions, to systematically work out appropriate answers, to interpret both negative and positive results, and ultimately present them in a comprehensible way.

With Jörg as my mentor, it almost felt like having a second supervisor. Thank you for every piece of advice and for keeping our working group close together. I am already looking forward to future birthdays or get-togethers in our usual hangout *Scotti's*.

I thank all of my (former) colleagues Ann-Kathrin, Anna, Daniel, Cynthia, Joanna, Lisa, Marc, Paul, Roman, Tessa, and Tobias for countless enlightening discussions and maintaining motivation. This also includes Simon, from whom I learned a lot during his short research stay. Most notably, I am grateful for having Christian and Robin around, who eventually became my co-authors and friends. Thank you for sharing ideas and ideals, for amazing me with your sharp focus and precision, and for always going the extra mile.

I am thankful Umberto welcomed me for an inspiring research stay at the university of Toulouse in early 2020, which led to years of insights and collaboration. I will always cherish the late-night discussions with Rachael about the depth of computational complexity theory, as well as our countless iterations of proofreading and shortening with Arianna. Jérôme's profound knowledge has always been a crucial resource to rely on, which often guided the way to novel and promising research directions.

There are far too many people to list, which have inspired me within the COMSOC community. Thank you for countless great talks, interesting conversations, valuable feedback (sometimes anonymously by reviewers), and for making me feel part of a kind group with many brilliant minds.

I thank everyone who was involved in my graduation program *Forschungskolleg Online-Partizipation* for funding me, as well as providing a stage for rewarding interdisciplinary discussions. Especially, I want to thank Martin and all of the (former) coordinators Lars, Tobias, Dennis, and Lena (who became my favorite next-door neighbor in the office).

I am grateful for all the friendly faces that make up the computer science department. Particularly, I want to thank Claudia for her diligent assistance in the background and always having an open ear, and Guido for keeping all systems running.

I am blessed with having three pairs of parents, which supported me over the years mentally, emotionally, and financially. My mother Bärbel taught me to stay focused and to work precisely. Without your trust and support in my undergraduate studies, I certainly would not have made it this far. My (step)father Dirk motivated me to keep going by taking one step at a time. My father Gerd got me interested in natural sciences and particularly computer science in the first place. Following your footsteps at an early age has led to an exciting life full of wonder. From my (step)mother Helga, I learned to find my own artistic style to present theoretical results appealingly. My parents-in-law Astrid and Joachim, who treat me like their own son, have been a wonderful addition to my family. Thank you for giving me a home throughout the tough times of the pandemic.

My older brother Felix has always been a huge inspiration throughout my life, sometimes paving the way for my own path. At some point, I grew up to be on a par with you, which made our discussions all the more fruitful. Being like-minded and having a similar academic background, you were my favorite and most appreciated person I could turn to, whenever I needed a professional outside perspective.

Finally, words alone cannot express my appreciation for my wife Vanessa's endless stream of support. You were my constant companion, always reminding me to eat enough and exercise (not only mentally), comforting and motivating me when I was struggling, and celebrating every little success with me. Thank you for being by my side.

Table of Contents

CHAPTER 1

INTRODUCTION

Since the very beginning of humankind, we are set in an environment, where we have to reach collective decisions on a regular basis. Often, we need to respect conflicting opinions, needs, and desires of complex individuals. While one can individually decide on whether to attend a wedding, doing chores or writing a thesis (assuming there are no opposing constraints), we need to reach a consensus when there are multiple deciders involved. On a large scale, those questions can relate to the election of a representative by a nation's population, agreeing on how to spend a limited public fund by local residents, or the verdict of guilt or innocence by a jury. But, we also encounter everyday decisions, that need to be taken jointly, such as choosing a restaurant with colleagues, finding a suitable movie to watch with friends, or picking a collectively enjoyed group activity.

In today's digital and interconnected world, where machines and algorithms often act autonomously on our behalf, it has become a natural part of our everyday lives to rely on collective decisions (often made for us instead of by us). Thus, it is crucial to understand and carefully choose suitable mechanisms, to make sure the final decisions are as beneficial as possible. Consider, for example, the mobility sector, which is becoming increasingly digitalized. Self-driving cars must convert a stream of (possibly contradicting) real-time data into a set of instructions carried out by the engine each split-second [9]. Especially if moral decisions are involved (e.g., if hitting a pedestrian can only be avoided by driving against a wall) the vehicle may become the judge of whom to endanger most [56]. Smart traffic light systems can reduce the average travel time for (most) drivers, by acting autonomously on the current traffic situation (e.g., by prioritizing busy lanes) [176]. To optimize the efficiency of public transport, a schedule can be planned based on the usual routes and travel times of passengers (which act as implicit preferences) [86]. Overall, we need to ensure that the resulting mechanisms are fair and efficient. A mathematically grounded valuation of the underlying principles can help us to derive reliable and verifiable implications, which ultimately allows for a profound discussion on which mechanism should be chosen for a specific use-case.

Luckily, questions revolving around (collective) decision-making processes and voting have been studied extensively in the research field of *social choice theory*. Its origin as a formal and mathematical research field dates back to the 18th century[1] and is usually associated with the pioneers Jean-Charles de Borda [35] and Marie Jean Antoine Nicolas Caritat, Marquis de Condorcet [55]. Over the years, social choice theory and related disciplines (such as game theory) have fascinated mathematicians, economists, philosophers, and eventually computer scientists. In the early stage of research on social choice theory, the focus was mostly on designing and analyzing specific aggregation mechanisms. The study was increasingly complemented by a more general, axiomatic approach, initiated by the groundbreaking impossibility result by Nobel laureate Kenneth Arrow [4] (stating there is no aggregation method satisfying a handful of reasonable properties). Finally, just by the beginning of this century, the subfield of *computational social choice* has emerged, by studying social choice theory through a computational lens, using techniques from computer science and artificial intelligence. In this context, *computational complexity theory* [3] — a subfield of theoretical computer science that aims to quantify the computational complexity associated with a problem in terms of resource requirements — turned out to be a useful toolbox for deriving key insights. For further reading on (computational) social choice, we refer to the books by McLean and Urken [129], Brandt, Conitzer, Endriss, Lang, and Procaccia [43], and Rothe [152].

Now, to embed the title of this thesis relatable to its contents, let us briefly motivate our primary focus and intended goals. As a foundation, we build on voting related research fields, where a set of participants with *individual* preferences takes part in a *collective* decision. Those fields are often studied independently and we explore three meaningful instances, namely *multiwinner elections*, *participatory budgeting*, and *judgment aggregation*. Deriving *individual* results for each research area on a *local level*, we continue to compile *global* implications to encourage the study on voting related fields as a *collective*.

Although the three aforementioned frameworks differ in motivation, use-case and notation, they share fundamental assumptions. In each framework, we consider binary valuations over a set of alternatives, which are aggregated from an individual level (i.e., the voters' approval ballots) to at least one set of winning alternatives as collective outcome. The aggregation process is done by using a voting rule, which (usually) must also abide a given constraint that models, which subsets of alternatives are feasible. Of course, a "good" voting rule should substantially reflect the voters' preferences. Structurally, the investigated subfields multiwinner elections, participatory budgeting, and judgment aggregation are mainly distinguishable by the type of constraint that is posed on the outcome to model feasibility.

In *multiwinner elections* with approval ballots [40, 81, 111], voters elect a fixed-size committee of candidates. Therefore, a suitable constraint for feasibility must require, that any outcome contains a predefined number of alternatives.

[1] Even though electoral processes and early forms of democracy can be found throughout history. For example, in 508 B.C.E., a participatory democracy was implemented during the Athenian revolution by the ancient Greeks [135]. In the 13th century, the Catalan philosopher Ramon Llull suggested a voting system based on pairwise comparison [94], which is similar to modern *Copeland* elections [80].

In *participatory budgeting* [156, 158, 8], residents specify which set of projects should be implemented by the municipality. Further, the actual implementation of a project is associated with a given cost, while the overall cost for realized projects should not exceed the available funds of a participatory budgeting campaign. Hence, a suitable budgeting constraint incorporates the projects' costs and the given budget limit.

Lastly, in *judgment aggregation* [124, 71, 19], a set of judges decides over the truthfulness of a set of (logically interconnected) allegations. Thus, the set of alternatives holds one truth value (true or false) for each allegation and any feasible outcome should render each allegation either true or false without resulting in a contradiction. Usually, we assume that the judges' opinions should also be consistent with the given logical constraint that models feasibility.

The similarity of the three studied frameworks allows us to investigate similar questions, whose answers can be derived using similar techniques. Overall, across all three research fields, the key results of this thesis revolve around five recurring research goals (stated explicitly in Chapter 3). That is, we use computational complexity theory and mathematical reasoning to uncover insights relating to *(i)* the axiomatic behavior of voting rules, *(ii)* the computational complexity of winner determination, *(iii)* the computational complexity of manipulative interference, *(iv)* relationships between independently studied voting rules, and *(v)* the potential of improvement for voting mechanisms by considering a slightly generalized ballot format.

Outline of this Thesis

The remainder of this thesis is organized as follows. In Chapter 2, we introduce the fundamental ideas and principles of computational complexity theory (see Section 2.1) and provide an overview of multiwinner elections (see Section 2.2), participatory budgeting (see Section 2.3), and judgment aggregation (see Section 2.4). In Chapter 3, we explore those three separate research fields through a more abstract lens, allowing us to motivate a total of five research goals at a higher level.

Then, each of the subsequent seven chapters is dedicated to one publication that was developed throughout graduation. In particular, we explore *"Irresolute Approval-based Budgeting"* [25] in Chapter 4, *"Complexity of Manipulative Interference in Participatory Budgeting"* [22] in Chapter 5, *"Time-Constrained Participatory Budgeting Under Uncertain Project Costs"* [24] in Chapter 6, *"Complexity of Sequential Rules in Judgment Aggregation"* [13, 14] in Chapter 7, *"Collective Combinatorial Optimisation as Judgment Aggregation"* [33, 34] in Chapter 8, *"Distortion in Attribute Approval Committee Elections"* [21] in Chapter 9, and *"Bounded Approval Ballots"* [23] in Chapter 10.

Finally, we conclude in Chapter 11, by *(i)* summarizing to what extend we were able to answer aspects of our initial research questions, *(ii)* deriving some additional results that arise from observing all individual contributions of this thesis as a whole, as well as *(iii)* discussing promising directions for future work.

CHAPTER 2

BACKGROUND

In this chapter, we formally introduce *computational complexity theory* (Section 2.1), as well as the three closely related research subfields of *computational social choice*,[2] namely *multiwinner elections* (Section 2.2), *participatory budgeting* (Section 2.3), and *judgment aggregation* (Section 2.4).

2.1 Computational Complexity Theory

In the age of digitalization we are accustomed to getting answers to seemingly complex questions in a matter of seconds. Using highly optimized algorithms on powerful machines in an interconnected world, we are able to render hyperrealistic three-dimensional models in real-time, simulate complex real-world phenomena (such as forecasting climate change), or update routes for navigation on the fly by identifying latest traffic bottlenecks via large-scale data analysis. More recently, artificial intelligence and learning-based algorithms are on the rise, outperforming us in a variety of human-like tasks, such as mastering the famous game of *Go* by playing against itself [157] or using high level natural language processing to either simulate human interactions [127] or generate high-quality images based on text prompts [60].

Admittedly, comprehensive advances in computation power and algorithm efficiency over the last decades rarely let us wonder, what we can not compute. Yet, by the 1930s, even long before commercially usable computers, it was discovered that there are problems, which are not *decidable* by any algorithm (i.e., a solution is not always computable). As most prominent example for such a problem, introduced and proven to be undecidable by Alan Turing [169] in 1936, the *halting problem* asks whether a given algorithm eventually terminates. Following this result, there has been a long history on the study of undecidable

[2]For further reading, see Brandt, Conitzer, Endriss, Lang, and Procaccia [43], as well as Rothe [152].

problems in the field of *computability theory* (for further reading, we refer to the textbook by Cooper [59]). Contrarily, moving to decidable problems (where a solution is always computable), *computational complexity theory* is concerned with classifying decidable problems in terms of resource requirements and efficiency.[3] By this day, there are still countless problems, for which it is still unknown, whether we are able to derive an answer *efficiently*, i.e., by using a deterministic algorithm in a reasonable amount of time.[4] To give an intuitive example of such a problem, consider the following everyday situation.

Example E1. *A smart art collector steps into a local antique shop, filled with a diverse and surprisingly valuable collection. Well prepared, the collector is aware of the appraised value for all pieces, which significantly exceeds the asking price. Having a limited budget on hand, what artworks should she purchase to maximize the potential return of her investment?*

Indeed, the problem illustrated in the above example, which is also known under the name KNAPSACK problem [103, 84], generally portraits a computationally *hard* problem (for which no known deterministic polynomial-time algorithm exists). To understand notions of *hardness* for a problem, the remainder of this section introduces all necessary concepts and is organized as follows. In Subsection 2.1.1, we formalize problem types for a structural study of so-called *decision problems*. In Subsection 2.1.2, we introduce *Turing machines* as an abstract computation model to quantify the complexity of a problem.[5] This allows us to group problems into so-called *complexity classes*, explored in Subsection 2.1.3. Finally, in Subsection 2.1.4, we discuss a notion of *hardness* for complexity classes, implying lower bounds on the resources necessary to solve a problem.

2.1.1 Problem Types and Instances

In theoretical computer science, we characterize problems by the description of a valid input (called *instance*) and an expected output type (declaring what kind of solution is expected). In Example E1, a valid instance is given by a finite collection of artworks, each associated with an appraised value and an asking price, and an available budget of funds to spend by the collector. Although problems are in some way unique, we can group various problems together by their expected output type. For *search problems* [1], as depicted in Example E1, the output to a given instance is a solution meeting the requested requirements (i.e., an affordable art selection that maximizes the expected return of investment). For *optimization problems* [108] the output is an optimal (usually maximum or minimum) value (e.g., the maximum achievable expected return of investment under the given bud-

[3] Although this chapter provides a short introduction to computational complexity theory, we refer to the textbooks by Arora and Barak [3], Papadimitriou [136], and Rothe [151] for an extensive overview.

[4] As a convention, we call functions (as well as problems) whose output can be computed by a deterministic algorithm in time polynomial in its input size *efficiently computable* or *tractable*.

[5] More precisely, the complexity of a problem refers to the complexity associated with the best performing algorithm that is able to solve the given problem.

get constraint). For *counting problems* [171], we are interested in the number of valid solutions (e.g., the number of possible ways to purchase artworks yielding a maximum expected return).

In this thesis, we mainly focus on *decision problems* [99, 169], where the output is always either YES or NO (e.g., describing whether there is an affordable selection of artworks, whose expected return surpasses a desired target value). Presented in a common notation, the decision variant of the KNAPSACK problem [84] (informally described in Example E1) can be formally defined as follows:

KNAPSACK	
Given:	Two finite lists of equal length containing positive integers to specify weights $W = (w_1, \dots, w_m) \in \mathbb{N}_+^m$ and utilities $U = (u_1, \dots, u_m) \in \mathbb{N}_+^m$ and two positive integers modeling a capacity and a target value $C, V \in \mathbb{N}_+$.
Question:	Is there a subset of items $S \subseteq [m]$ with $\sum_{i \in S} w_i \leq C$ and $\sum_{i \in S} u_i \geq V$?[6]

For a decision problem A, we say that $\mathcal{I}$ is an instance of A, if and only if its contents do abide the specified input requirements. Further, we say $\mathcal{I}$ is a YES-instance for A (also written as $\mathcal{I} \in A$), if and only if the answer to the question in A is YES, given the input $\mathcal{I}$. Analogously, we refer to an instance $\mathcal{I} \notin A$ as NO-instance for A.[7] As a decision problem A can be characterized by its YES-instances, we sometimes write A as *formal language*.[8] That is, assuming a suitable encoding of an instance $\mathcal{I}$ as a finite string (called word) over a finite alphabet consisting of unique symbols, A can be interpreted as the (possibly infinite) collection of all words encoding YES-instances, i.e.,

$$A = \{\mathcal{I} \mid \mathcal{I} \text{ is a YES-instance for } A\}.$$

Exploiting the duality of the output of decision problems, let us denote by $\overline{A}$ the *complement* of a decision problem A, containing all NO-instances. That is,

$$\overline{A} = \{\mathcal{I} \mid \mathcal{I} \text{ is a NO-instance for } A\}.$$

Extrapolating this principle of duality from instances to problems, we can group all kinds of problems into formal languages for a more structured analysis. For example, we might define a *class* of problems $\mathcal{C}$, which contains all decidable decision problems. In a similar way, for any class of decision problems $\mathcal{C}$, its so-called *co-class* co$\mathcal{C}$ contains all complement problems for $\mathcal{C}$, i.e.,

$$\text{co}\mathcal{C} = \{\overline{A} \mid A \in \mathcal{C}\}.$$

In computational complexity theory, a classification of problems is usually done by grouping problems with a similar complexity (i.e.., time and/or resource requirements necessary

[6] For integers i, j with $i \leq j$, let $[i, j] = \{i, i+1, \dots, j\}$ and $[i] = [1, i]$ denote interval sets.

[7] For example, it is easy to verify that $((1), (1), 1, 1)$ is a YES-instance for the KNAPSACK problem, while $((1, 2), (2, 3), 3, 6)$ is a NO-instance, and $((1, 2), (2), 4)$ is not an instance at all, as the lists of integers do not have matching length and one parameter is missing.

[8] Originated from the work by Chomsky [52]. For further reading, see the textbook by Harrison [95].

to find a solution) into so-called *complexity classes*. As an intuitive informal example, the complexity class P is the class of all decision problems, which can be decided by a deterministic algorithm whose number of steps remains polynomial in the input size. A structured mathematical foundation to classify problems by time and resource requirements, originated by Hartmanis and Stearns [96], relies on *Turing machines*. Turing machines, introduced by and named after Alan Turing [169], are a powerful computational model to quantify the complexity of a given problem. Following the Church-Turing thesis [53, 169], which is generally assumed to be true, any intuitively computable function (and thus any executable algorithm) can be translated into a corresponding Turing machine and vice versa. We formally introduce Turing machines in the upcoming subsection, to be able to properly define all complexity classes relevant to this thesis in Subsection 2.1.3.

2.1.2 Turing Machines

To compute an output from a given input, we use an abstract computational model introduced by Turing [169], namely *Turing machines*. A Turing machine is a simple abstract machine equipped with the following four basic components. An infinite *memory tape* is divided into single cells that can hold up to one symbol from a finite alphabet. A *state register* holds the current state of the machine, chosen from a finite set of predefined states. A *head*, which is always pointed at exactly one cell of the memory tape, is able to read and overwrite its current cell's content and optionally move to an adjacent cell. Finally, a finite set of *transition rules* dictates the action of the head and the next state of the machine, based on the current content of the state register and the symbol read at the current position of the head.[9]

To perform a computation, a given Turing machine starts from an initial *configuration*,[10] where an input word is written on the memory tape with the head pointing at the leftmost symbol, while the state register is holding an initial state. Then, the Turing machine does one computational step at a time according to its transition rules, until eventually the machine holds and accepts the input if and only if a clearly marked accepting state is reached. For simplicity, we assume that, if there is no transition rule explicitly defined for a symbol and a state, the machine stays in this position forever (i.e., the current symbol, the state register and the head's position will remain unchanged). Moreover, if an accepting state is unreachable from a configuration, the machine will loop forever.

We distinguish between *deterministic Turing machines* (DTMs), where there is exactly one transition rule for each suitable pair (of the current state and the most recently read symbol) and *non-deterministic Turing machines* (NTMs), where there can be multiple transition rules for a configuration. For the latter, the machine branches its computational

[9]Note that (along with a predefined set of states and transition rules) the distinct description of an explicit Turing machine requires further specifications, such as an initial state and a (set of) accepting state(s).

[10]A configuration can be interpreted as a snapshot of the machine at a given point of time, capturing the contents of the memory tape, the head's position and the state of the machine.

path into multiple transitions in parallel and we assume an input is accepted by a given NTM if and only if there is at least one path resulting in an accepting state.

In case an accepting state is reached, we can infer an output from the contents of the memory tape, allowing us to find answers for functional problems, optimization problems, or search problems. Analogously, for counting problems the correct answer may be derived from an NTM's number of accepting paths. Yet, for this thesis (concerned with decision problems) it is sufficient to interpret Turing machines as *decider model* to determine whether a given input word results in an accepting state.

For a Turing machine M, we denote by $L(M)$ the formal language, containing all inputs that result in an accepting state. That is, expecting an input word from a predefined alphabet Σ,[11]

$$L(M) = \{w \in \Sigma^* \mid M \text{ reaches an accepting state on input } w\}.$$

Assuming a suitable encoding of a (decidable) decision problem A, this allows us to express A by $L(M) = A$ for a matching Turing machine M accepting A. In terms of efficiency, there might be infinitely many machines accepting A. Some of which might compute the result using fewer computational steps, require less memory, or only use deterministic transitions. To explore the resource requirements for a Turing machine to model a decision problem, let us briefly introduce *time complexity* [96] and *oracle Turing machines* [170], allowing us to properly define all complexity classes that are relevant to this thesis in the upcoming subsection.

Time Complexity

The problems encountered in this thesis mostly rely on *time complexity*, measured by the number of computational steps required to decide on the acceptance of a given input (with respect to its size). Formally, when considering a Turing machine M along with an input word w (or analogously a decision problem A with a suitable instance $\mathcal{I}$), we define its length $|w|$ (or $|\mathcal{I}|$) as the size of the input. Note that for an alphabet Σ consisting of $|\Sigma| = m$ symbols, there are m^n potential inputs with size $n \in \mathbb{N}_0$.[12] Given an algorithm or, in particular, a Turing machine M along with an input w, we quantify time by the number of computational steps necessary to decide whether $w \in L(M)$ holds.

To measure the worst-case run-time of a Turing machine, we make use of the Bachmann-Landau notation [114]. A function $f\colon \mathbb{N}_0 \to \mathbb{N}_0$ is bounded asymptotically upwards by a function $g\colon \mathbb{N}_0 \to \mathbb{N}_0$, written $f \in \mathcal{O}(g)$, if there exists a constant $c \in \mathbb{R}_+$ and a sufficiently large value $n_0 \in \mathbb{N}_0$, such that for all $n \geq n_0$ it holds that $f(n) \leq c \cdot g(n)$. In this thesis, we mainly distinguish between constant functions ($\mathcal{O}(1)$), logarithmic functions ($\mathcal{O}(\log(n))$), linear functions ($\mathcal{O}(n)$), polynomial functions ($\mathcal{O}(n^{\mathcal{O}(1)})$), and exponential functions ($\mathcal{O}(c^n)$ for $c > 1$).

[11] Σ^* refers to the set of words, that can be formed using a finite number of letters from alphabet Σ.

[12] We usually assume binary encoding over the fixed alphabet $\Sigma = \{0, 1\}$.

Transferring asymptotic bounds to the run-time of a Turing machine M, we say that M has a run-time bounded by a function g, if any accepted input $w \in \Sigma^*$ of size $|w| = n$ can be decided in at most $\mathcal{O}(g(n))$ computational steps. Hence, a *polynomial-time bounded Turing machine* M, given any (accepted) input $w \in L(M)$ of length $|w| = n$, always terminates after at most $\mathcal{O}(n^{\mathcal{O}(1)})$ steps (or loops forever in case $w \notin L(M)$).

Although not focus of this thesis, we might similarly investigate the *space complexity* of a Turing machine by measuring the number of memory cells required for a successful computation, as suggested by Stearns, Hartmanis, and Lewis [163]. Observe that space is limited by time, as we require one computational step to manipulate the content of a cell.

Oracle Turing Machines

An *oracle Turing machine* M is equipped with an additional memory tape (called *oracle tape*) and a so-called *oracle* for a predefined decision problem A, written M^A. By querying its A-oracle, the Turing machine is able to decide an instance for A in one computational step. More precisely, for any precomputed word w, M^A can verify at any time whether $w \in A$ holds and write the answer (i.e., zero or one) onto a cell of the oracle tape. We extend this definition by considering a class $\mathcal{C}$ of decision problems as an oracle instead of a single problem.[13] Then, $M^{\mathcal{C}}$ may query its $\mathcal{C}$-oracle to decide instances for a fixed problem in $\mathcal{C}$, inducing the language $L(M^{\mathcal{C}}) = \bigcup_{A \in \mathcal{C}} L(M^A)$.

For a more evolved study on the usage of a given oracle, we may interpret the number of queries required for a successful computation as a resource requirement. If the number of allowed oracle queries is limited by a fixed constant $k \in \mathbb{N}_+$, we write $M^{\mathcal{C}[k]}$. If the number of queries must be logarithmic in the input size, we write $M^{\mathcal{C}[\log]}$. In a related definition (see Kadin [101]), the oracle can only be accessed once to decide multiple instances for a problem in $\mathcal{C}$ in parallel at any time of the computation, denoted by $M^{\mathcal{C}}_{||}$. Note that k queries posed sequentially (modeling one path in a binary decision tree) can be simulated by posing $2^k - 1$ queries in parallel [151].

2.1.3 Complexity Classes

Finally, we are set up to formally introduce all *complexity classes* encountered in this thesis. As we see at the end of this subsection, the considered classes are all contained in the second level of the *polynomial-time hierarchy* [130, 164], some of which are contained in the *Boolean hierarchy* over NP [47, 48]. In the following, we introduce each of the relevant complexity classes separately, before discussing how those classes are related.

To give a more intuitive understanding, we complement formal definitions with canonical decision problems, falling exactly into the respective classes. To do so, let us take a short detour and define *propositional logic* first, to subsequently provide variations of

[13]Recall that a class $\mathcal{C}$ simply is a (possibly infinite) collection of problems.

the famous SATISFIABILITY problem [58, 103] (usually denoted by SAT) for each of the relevant complexity classes.

Definition (Propositional Logic)**.** *In the language of propositional logic, we consider a set of atomic propositions $X = \{x_1, \ldots, x_n\}$, which can hold exactly one of two truth values, i.e., $x_i \in \{0,1\}$ for all $i \in [n]$ (where $x_i = 1$ implies that x_i evaluates to* TRUE *and $x_i = 0$ implies that x_i evaluates to* FALSE*). Using the connectives $\neg$ (negation), $\wedge$ (conjunction), $\vee$ (disjunction), $\rightarrow$ (implication), and $\leftrightarrow$ (equivalence), as well as the constant propositions 0 and 1, we can construct more evolved propositional formulas inductively as follows. Given a truth assignment $x \in \{0,1\}^n$ for the atomic propositions X and two propositional formulas α and β, the following holds.*

$$\begin{array}{lll} \varphi = \neg\alpha & \textit{is a propositional formula with} & \varphi(x) = 1 - \alpha(x). \\ \varphi = \alpha \wedge \beta & \textit{is a propositional formula with} & \varphi(x) = \min(\alpha(x), \beta(x)). \\ \varphi = \alpha \vee \beta & \textit{is a propositional formula with} & \varphi(x) = \max(\alpha(x), \beta(x)). \end{array}$$

Further, $\alpha \rightarrow \beta$ is shorthand for $(\neg\alpha) \vee \beta$ and $\alpha \leftrightarrow \beta$ is short for $(\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha)$. Finally, a truth assignment x satisfies a propositional formula φ if and only if $\varphi(x) = 1$.

For a proportional formula φ over a set of atomic propositions X with $|X| = n$, let $L(\varphi) = \{x \in \{0,1\}^n \mid \varphi(x) = 1\}$ denote the formal language, that consists of all satisfying truth assignments for φ. In this subsection, we present all canonical decision problems as questions one might ask about the (not explicitly given) set of satisfying assignments $L(\varphi)$, induced by a given formula φ. This allows for a unified perspective and for a more profound understanding of the specific factors that contribute the complexity of a given problem. We illustrate the respective problems compactly in Figure 2.1 (respective complexity classes are defined throughout the remainder of this subsection) and the relationships between relevant complexity classes in Figure 2.2.

The Class P

The complexity class P contains all decision problems that can be decided in deterministic polynomial time with respect to the size of an input instance. Formally, using Turing machines as a computational model for P, that is

$$\mathrm{P} = \{A \mid \text{There is a polynomial-time bounded DTM } M \text{ with } L(M) = A\}.$$

Note that P is closed under complement, i.e., $\mathrm{P} = \mathrm{coP}$.

A typical example for a problem in P is to decide whether a given propositional formula φ evaluates to TRUE for a given truth assignment $x \in \{0,1\}^n$ of the n atomic variables in φ. Formally, the question, whether $x \in L(\varphi)$ holds, lies in P. Indeed, it is easy to verify, whether x satisfies φ, by sequentially resolving the connectives in φ.

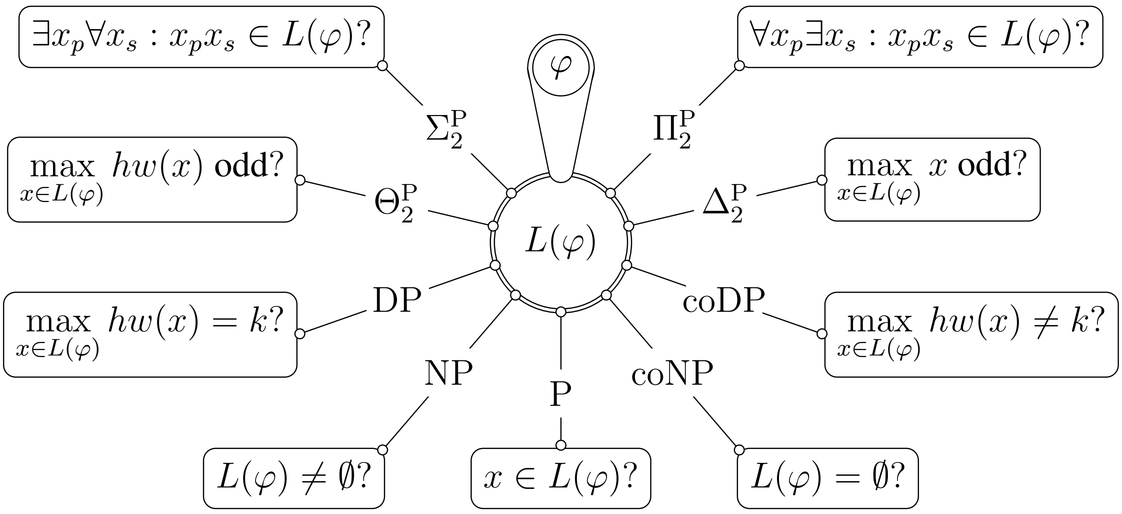


Figure 2.1. Collection of decision problems for different complexity classes, formulated for a given propositional formula φ. Note that x is explicitly given for the problem in P; k is a given variable for the problems in DP and coDP; and for the problems in Σ_2^{P} and Π_2^{P}, $x_p \in \{0,1\}^k$ (respectively $x_s \in \{0,1\}^{n-k}$) has a predefined length k, specified by an instance. Lastly, $hw(x) = \sum_{i \in [|x|]} x_i$ refers to the hamming weight of a binary string x.

The Classes NP and coNP

The complexity class NP contains all decision problems, whose YES-instances can be verified in deterministic polynomial time. For verification, we can use a *certificate* of polynomial length (called *witness*). As a computation model for the class NP, we consider non-deterministic Turing machines with a polynomial run-time, i.e.,

$$\mathrm{NP} = \{A \mid \text{There is a polynomial-time bounded NTM } M \text{ with } L(M) = A\}.$$

To link both definitions closer together, note that any branch resulting in an accepting state (in polynomial time) can be interpreted as a universal witness.

The canonical problem for the class NP is the SATISFIABILITY problem [58, 103] (SAT), where the task is to decide, whether a given propositional formula φ is satisfiable (by at least one truth assignment), i.e., whether $L(\varphi) \neq \emptyset$ holds. It is easy to see that this problem belongs to the class NP, as we can use any satisfying truth assignment $x \in L(\varphi)$ as a witness to verify that φ is satisfiable. Note that we are not able to provide a suitable witness in case φ is unsatisfiable.

In turn, the complexity class coNP contains those problems, whose NO-instance can be verified in deterministic polynomial time. We may analogously formulate the complementing problem for SAT, namely $\overline{\text{SAT}}$, as follows. Given a propositional formula φ, the question is whether φ is unsatisfiable, i.e., whether $L(\varphi) = \emptyset$ holds. Similarly, a NO-instance can be identified by any satisfying truth assignment $x \in L(\varphi)$ as witness.

The Classes DP and coDP

Any decision problem in the complexity class DP, formally introduced by Papadimitriou and Yannakakis [140], may be seen as intersection of an NP-problem and a coNP-problem. In particular most DP problems can be split into two parts: An NP-question and a coNP-question. The answer to a given DP-instance is YES if and only if the answers to both, NP and coNP, questions are YES. Formally, that is

$$\mathrm{DP} = \{L_1 \cap L_2 \mid L_1 \in \mathrm{NP} \text{ and } L_2 \in \mathrm{coNP}\}.$$

Formulating a suitable problem for DP,[14] we may ask for a given satisfiable propositional formula φ, whether the largest number of positive literals contained in a satisfying assignment is equal to a given parameter $k \in \mathbb{N}_0$. Formally, if $hw(x) = \sum_{i\in[|x|]} x_i$ refers to the *hamming weight* of x, we ask whether there exists an $x \in L(\varphi)$ with $hw(x) \geq k$ and there exists no $y \in L(\varphi)$ with $hw(y) > k$ (i.e., $y \in L(\varphi)$ implies $hw(y) \leq k$).

In turn, problems in coDP may be split in a similar way, where the answer to a given coDP-instance is YES if and only if at least one of the two posed NP- and coNP-questions returns YES, i.e.,

$$\mathrm{coDP} = \{L_1 \cup L_2 \mid L_1 \in \mathrm{NP} \text{ and } L_2 \in \mathrm{coNP}\}.$$

For an analogous coDP problem, we may ask, whether the above does not hold. Presenting the resulting NP-question first, we ask if there exists an $y \in L(\varphi)$ with $hw(y) > k$ or there exists no $x \in L(\varphi)$ with $hw(x) \geq k$ (i.e., $x \in L(\varphi)$ implies $hw(x) < k$).

The Classes Θ_2^{P} and Δ_2^{P}

The class $\Theta_2^{\mathrm{P}} = \mathrm{P}^{\mathrm{NP[log]}}$, originally introduced by Papadimitriou and Zachos [138] (and respectively the class $\Delta_2^{\mathrm{P}} = \mathrm{P}^{\mathrm{NP}}$ [164]), contains exactly those problems that can be decided in deterministic polynomial time, while querying an NP-oracle a logarithmic (respective polynomial) number of times. Both classes are closed under complement, i.e., $\Theta_2^{\mathrm{P}} = \mathrm{co}\Theta_2^{\mathrm{P}}$ and $\Delta_2^{\mathrm{P}} = \mathrm{co}\Delta_2^{\mathrm{P}}$.

Natural problems in the classes Θ_2^{P} and Δ_2^{P} are decision problems where a (computationally hard) optimization problem has to be solved first. Investigating a continuous search space for the value to optimize, we may use binary search, where each search step requires solving an NP-problem. Overall, if the space of possible values to optimize is bounded by a polynomial (respectively, is exponential), binary search requires solving a logarithmic (polynomial) number of NP-queries, resulting in membership to Θ_2^{P} (or Δ_2^{P}).

Building on the results by Krentel [108], a canonical problem for Θ_2^{P} is to decide for a propositional formula φ, whether a satisfying assignment with the highest number of positive literals contains an odd number of positive literals. Formally, asking whether $\max_{x\in L(\varphi)} hw(x)$ is odd is in Θ_2^{P}. For Δ_2^{P}, we may ask whether the lexicographically maximum satisfying assignment is odd, i.e., whether $\max_{x\in L(\varphi)} x$ is odd.

[14] For related problems, see Papadimitriou and Yannakakis [140] and Papadimitriou and Wolfe [137].

The Classes Σ_2^{P} and Π_2^{P}

The class $\Sigma_2^{\mathrm{P}} = \mathrm{NP}^{\mathrm{NP}}$ contains all decision problems, whose YES-instances can be decided by a non-deterministic algorithm with access to an NP-oracle in polynomial time. Hence, an algorithmic model for Σ_2^{P} are polynomial-time bounded non-deterministic Turing machines equipped with an NP-oracle. Its co-class $\Pi_2^{\mathrm{P}} = \mathrm{co}\Sigma_2^{\mathrm{P}} = \mathrm{coNP}^{\mathrm{NP}}$ respectively contains the problems, whose NO-instances can be decided by such an algorithm.

As canonical problems for both classes, consider a propositional formula φ over propositional variables $X = \{x_1, \dots, x_n\}$, partitioned by a given parameter $k \in [n]$ into two sets $X_p = \{x_1, \dots, x_k\}$ and $X_s = \{x_{k+1}, \dots, x_n\}$ to model prefix and suffix. A question in Σ_2^{P} (also known as QUANTIFIED BOOLEAN FORMULA [164, 175]) is, whether there exists a prefix $x_p \in \{0,1\}^k$ (i.e., a partial truth assignment over X_p), such that all extensions of x_p by a suffix $x_s \in \{0,1\}^{n-k}$ (i.e., an assignment over X_s) yield a satisfying assignment $x_p x_s \in L(\varphi)$. In contrast, a typical Π_2^{P}-question is whether every prefix $x_p \in \{0,1\}^k$ can be extended by a suffix $x_s \in \{0,1\}^{n-k}$ to a satisfying assignment $x_p x_s \in L(\varphi)$.

Hierarchies over NP

For completeness, let us conclude this subsection by presenting two inductive definitions: For the *Boolean hierarchy* over NP (denoted by $\mathrm{BH}(\mathrm{NP})$), formally defined by Cai et al. [47, 48], and the *polynomial-time hierarchy* (denoted by PH), introduced by Meyer and Stockmeyer [130, 164]. This allows us to embed all previously defined complexity classes into a larger context.

The lowest levels of the Boolean hierarchy $\mathrm{BH}(\mathrm{NP})$ are defined as $\mathrm{BH}_0(\mathrm{NP}) = \mathrm{P}$ and $\mathrm{BH}_1(\mathrm{NP}) = \mathrm{NP}$. For all $k \geq 1$, the remaining levels can be defined inductively as

$$\begin{aligned}
\mathrm{BH}_{2k}(\mathrm{NP}) &= \{L_1 \cap L_2 \mid L_1 \in \mathrm{coNP} \text{ and } L_2 \in \mathrm{BH}_{2k-1}(\mathrm{NP})\} \text{ and} \\
\mathrm{BH}_{2k+1}(\mathrm{NP}) &= \{L_1 \cup L_2 \mid L_1 \in \mathrm{NP} \quad \text{and} \quad L_2 \in \mathrm{BH}_{2k}(\mathrm{NP})\}.
\end{aligned}$$

Overall, the Boolean hierarchy $\mathrm{BH}(\mathrm{NP})$ contains the union of all complexity classes described above, i.e.,

$$\mathrm{BH}(\mathrm{NP}) = \bigcup_{k=0}^{\infty} \mathrm{BH}_k(\mathrm{NP}).$$

The polynomial-time hierarchy PH can be defined inductively in a similar way. For the bottom level, it holds that $\mathrm{P} = \Delta_0^{\mathrm{P}} = \Sigma_0^{\mathrm{P}} = \Pi_0^{\mathrm{P}}$. From there it is sufficient to define the $(i+1)$-th level with $i \geq 0$. That is, $\Delta_{i+1}^{\mathrm{P}} = \mathrm{P}^{\Sigma_i^{\mathrm{P}}} = \mathrm{P}^{\Pi_i^{\mathrm{P}}}$,[15] $\Sigma_{i+1}^{\mathrm{P}} = \mathrm{NP}^{\Sigma_i^{\mathrm{P}}} = \mathrm{NP}^{\Pi_i^{\mathrm{P}}}$ and $\Pi_{i+1}^{\mathrm{P}} = \mathrm{coNP}^{\Sigma_i^{\mathrm{P}}} = \mathrm{coNP}^{\Pi_i^{\mathrm{P}}}$. As an important remark, note that a polynomial-time

[15] As analogues for $\Delta_{i+1}^{\mathrm{P}}$ with a bounded number oracle queries, we can define infinitely many complexity classes $\mathrm{P}^{\Sigma_i^{\mathrm{P}}[1]} \subseteq \mathrm{P}^{\Sigma_i^{\mathrm{P}}[2]} \subseteq \dots \subseteq \mathrm{P}^{\Sigma_i^{\mathrm{P}}[\log]} = \Theta_{i+1}^{\mathrm{P}}$ with $\Sigma_i^{\mathrm{P}} \cup \Pi_i^{\mathrm{P}} \subseteq \mathrm{P}^{\Sigma_i^{\mathrm{P}}[1]}$ and $\Theta_i^{\mathrm{P}} \subseteq \Delta_i^{\mathrm{P}}$. An equivalent definition of $\Theta_{i+1}^{\mathrm{P}} = \mathrm{P}_{||}^{\Sigma_i^{\mathrm{P}}}$ allows for a polynomial number of oracle queries in parallel [97, 46].

bounded Turing machine equipped with a P-oracle can be simulated by the underlying machine directly, deriving the following classes for the first level: $\Delta_1^{\mathrm{P}} = \mathrm{P}^{\mathrm{P}} = \mathrm{P}$, $\Sigma_1^{\mathrm{P}} = \mathrm{NP}^{\mathrm{P}} = \mathrm{NP}$ and $\Pi_1^{\mathrm{P}} = \mathrm{coNP}^{\mathrm{P}} = \mathrm{coNP}$. Finally, the polynomial-time hierarchy PH is defined as union of all its contained levels, i.e.,

$$\mathrm{PH} = \bigcup_{i=0}^{\infty} \Delta_i^{\mathrm{P}} = \bigcup_{i=0}^{\infty} \Sigma_i^{\mathrm{P}} = \bigcup_{i=0}^{\infty} \Pi_i^{\mathrm{P}}.$$

Figure 2.2 illustrates relationships between complexity classes in the polynomial-time hierarchy. Classes further to the left are contained in classes further to the right, showcasing how $\mathrm{BH(NP)}$ is contained in the second level of PH.

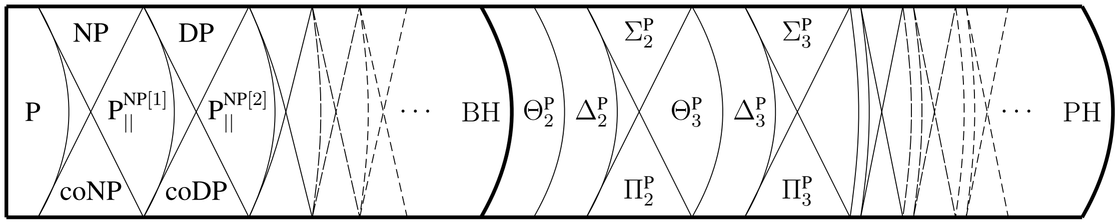


Figure 2.2. Illustration of $\mathrm{BH(NP)}$ contained inside the second level of PH. Complexity classes further to the left are contained in complexity classes further to the right.

2.1.4 Reducibility, Hardness, and Completeness

To this point, we have only investigated complexity classes from an upper bound perspective. That is, showing that a problem A is contained in a complexity class $\mathcal{C}$ implies that A can be decided by the computational model defined for $\mathcal{C}$. Yet, it remains unclear whether we can solve A using less resources, i.e., whether $A \in \mathcal{C}' \subset \mathcal{C}$ holds for a class $\mathcal{C}'$ with significantly lower complexity requirements. For example, showing $\textsc{Sat} \in \mathrm{NP}$ reveals that any $\textsc{Sat}$ instance can be solved by a polynomial-time bounded non-deterministic Turing machine, without giving an indication whether non-determinism is a necessary requirement for solving (i.e., whether even $\textsc{Sat} \in \mathrm{P}$ could hold). To establish lower bounds on the computational complexity of a problem, we consider so-called *reductions* to show that a problem B is *at least as hard* to decide as all problems of a given complexity class $\mathcal{C}$.

Let us briefly motivate the concept of reductions and embed it into the larger context of theoretical computer science. Reductions are a relatively simple, yet powerful tool to derive valuable insights about computational aspects of a given problem. Informally, instead of designing an algorithm for a given problem from scratch, a reduction relies on algorithmically solving a problem A by building on the knowledge of how to solve a problem B. In particular, this is done by specifying how to translate any instance of A into an instance of B, such that YES- and NO-instances are preserved. Without any further restrictions posed on the underlying reduction, let us illustrate what implications reducing A to B hold in the context of computability theory. If B is decidable, so must be A, as by construction we can solve A through B. By contraposition, if A is undecidable, we can deduce that B is undecidable, too.

To meaningfully extend this idea into the context of computational complexity theory, we pose restrictions on the available resources (e.g., time and space) of a reduction (conceptually similar to complexity classes). Assume we have found a reduction from A to B, and B belongs to a complexity class $\mathcal{C}$. If the reduction in question is *sufficiently efficient* (i.e., has an insignificant complexity in contrast to the complexity required to solve problems in $\mathcal{C}$), then $A \in \mathcal{C}$ follows immediately by construction. Vice versa, if we know that A is *hard* for $\mathcal{C}$ (i.e., cannot be solved using significantly less resources than available by the computational model for $\mathcal{C}$), then B must be at least as hard to solve as A.

Although there are manifold reduction types,[16] for this thesis we only consider *polynomial-time many-one reductions* (for decision problems),[17] defined as follows.

> **Definition** (Polynomial-time many-one reduction)**.** *Let A and B be two decision problems with a suitable binary encoding for respective instances. Then A is* polynomial-time many-one reducible *to B, written as $A \leq_m^p B$, if there is a polynomial-time computable function (with respect to its input's size) $f\colon \{0,1\}^* \to \{0,1\}^*$, such that for all $\mathcal{I} \in A$ it holds that*
>
> $$\mathcal{I} \in A \Leftrightarrow f(\mathcal{I}) \in B.$$

By design, it is easy to verify that $\leq_m^p$ is a reflexive and transitive relation. Indeed, for any decision problem, reflexivity follows by using the identity function (i.e., $f(\mathcal{I}) = \mathcal{I}$), while transitivity follows from nesting two polynomial-time computable functions (i.e., $f(\mathcal{I}) = f_1(f_2(\mathcal{I}))$) and the fact that the number of computational steps for the resulting composite function remains asymptotically bounded upwards by a polynomial.

Having a fixed reduction type in place, namely polynomial-time many-one reductions, let us discuss for which complexity classes, $\leq_m^p$ is *sufficiently efficient* to actually derive meaningful implications via reductions. Informally, we call a complexity class $\mathcal{C}$ *closed* under $\leq_m^p$, if the resources used by any such reduction cannot significantly impact (i.e., exceed) the resources available by $\mathcal{C}$. In particular, if $A \leq_m^p B$ and $\mathcal{C}$ is closed under $\leq_m^p$, then $B \in \mathcal{C}$ implies $A \in \mathcal{C}$. This allows us to derive implications about the lower bound complexity of a problem, i.e., the minimum resource requirement to solve a problem. A problem B is called $\mathcal{C}$*-hard*, if $\mathcal{C}$ is closed under $\leq_m^p$ and every problem $A \in \mathcal{C}$ is reducible to B, i.e., $A \leq_m^p B$.[18] Informally, this indicates that B is computationally *at least as hard to solve* as any other problem in $\mathcal{C}$. A $\mathcal{C}$-hard problem B is called $\mathcal{C}$*-complete*, if it additionally holds that $B \in \mathcal{C}$.

[16] Prominent reductions being Turing [57], metric [108], truth-table [112], or log-space reductions [113].
[17] Also known as Karp reductions, following the work by Karp [103].
[18] By transitivity of $\leq_m^p$, $\mathcal{C}$-hardness follows by showing $A \leq_m^p B$ for only one $\mathcal{C}$-hard problem A.

It turns out, that polynomial-time many-one reductions ($\leq_m^p$) are sufficiently efficient to derive closure for all complexity classes in the scope of this thesis. In particular, for any class $\mathcal{C}$ with $\mathrm{P} \subseteq \mathcal{C}$, it holds that $\mathcal{C}$ is closed under polynomial-time many-one reductions. To this day, the exact inclusion relationships between countless complexity classes remain an open problem. As most prominent representative, the question whether $\mathrm{P} \subseteq \mathrm{NP}$ is either a real inclusion or an identity is among the seven *Millennium Prize Problems* [58]. That means, although an NP-complete problem B might be solvable by a deterministic polynomial-time algorithm (effectively proving $\mathrm{P} = \mathrm{NP}$), there is no known, efficient algorithm available today to effectively solve B. In case $\mathrm{P} \subsetneq \mathrm{NP}$, there will never be such an algorithm.

We conclude this section with various references to related literature on complete problems for the complexity classes listed in Subsection 2.1.3. First of all, note that any presented variation of the satisfiability problem, depicted in Figure 2.1, is complete for its respective complexity class.[19] As a groundbreaking starting point, both, Cook [57] and Levin [123], independently demonstrated NP-completeness for SAT by reducing the description of any non-deterministic Turing machine along with an input to a proposition formula of polynomial size. Having a canonical base problem to reduce from in place, Karp [103] used polynomial-time many-one reductions to prove NP-completeness for 20 additional problems. An extensive list of many NP-complete problems, can be found in the textbook by Garey and Johnson [84]. Schaefer and Umans [155] provided a similar list for the second and third levels of the polynomial-time hierarchy. Papadimitriou [139] was the first one to present a natural complete problem for Δ_2^{P}, namely asking whether an optimal route for a *traveling salesperson* problem is a unique solution. Krentel [108] provided a list of several complete problems for Θ_2^{P} and Δ_2^{P} and showed how to derive DP-hard problems from related optimization problems. Lastly, Wagner [172] and Lukasiewicz and Malizia [128] showed how to construct Θ_2^{P}-hard problems from NP-complete problems. Notably, Hemaspaandra, Hemaspaandra, and Rothe [98] exploited the result by Wagner, to detect the first natural Θ_2^{P}-complete problem in the context of computational social choice, namely whether a distinct candidate wins a *Dodgson election* (named after Charles Dodgson [67], better known under his pen name Lewis Carroll).

[19] Completeness for the introduced problems in NP, Δ_2^{P}, and Σ_2^{P} follow from cited references. For DP and Θ_2^{P}, let us provide a simple reduction. For every graph $G = (V, E)$, we can find a compact formula φ in 2-CNF, such that $L(\varphi)$ models exactly the set of all cliques [84] appearing in G (by adding clause $(\neg x \vee \neg y)$ for every (missing) edge $\{x, y\} \notin E$). Deciding whether the size of a largest clique is equal to k (respectively is odd) is DP-hard [140] (Θ_2^{P}-hard [108]). Completeness for co-classes follows immediately.

2.2 Multiwinner Elections

In multiwinner elections, the goal is to elect a set of candidates based on individual voters' preferences. Applied to real-world scenarios, the set of elected candidates may be a committee of experts, a selection of talks at a conference or finding a set of suitable appointments for a recurring meeting. Outside the scope of this thesis, a vast amount of literature considers elections with ordinal (or cardinal) preferences [20, 177], which sometimes are assumed to abide a predefined structure [70]. In this thesis, we mainly focus on dichotomous preferences using approval ballots (formally introduced in the upcoming subsection) and two extensions (introduced in Subsection 2.2.2).

2.2.1 Approval-based Committee Elections

In this subsection we, focus on multiwinner elections with two popular restrictions, widely considered throughout related literature: *(i)* we limit feasible outcomes to be fixed-size committees, and *(ii)* we assume the voters' preferences are cast using approval ballots (where each voter submits a set of preferred candidates).[20]

Formally, in multiwinner elections, we are given a set of m *alternatives* (also referred to as *candidates*) $C = \{c_1, \dots, c_m\}$ and a set of n *agents* (called *voters*) $V = \{v_1, \dots, v_n\}$, where each voter $v_i \in V$ casts an *approval ballot* $B_i \subseteq C$ to express her individual *preference* over the set of candidates. Then each multiwinner election E is characterized by a pair $E = (C, V)$ and the collection $\mathcal{E}$ contains all possible elections $E \in \mathcal{E}$. For a positive integer $k \in \mathbb{N}_+$, let $\mathcal{P}_k(C) = \{W \subseteq C \mid |W| = k\}$ be the set of all k-*committees*. Finally, an *(approval-based committee) voting rule* F maps an election $E \in \mathcal{E}$ along with an integer $k \in \mathbb{N}_+$ to a non-empty set of winning k-committees $F(E, k) \subseteq \mathcal{P}_k(C)$.[21]For illustration, consider the following example.

Example E2. *For a long-distance flight, an airline offers a movie streaming service for their passengers, which is included in the ticket price. The airline pays per movie (instead of per stream) and each customer is asked to select those movies, she might enjoy. To reduce cost for the airline, only three movies will be available throughout the flight. Formally, let* $C = \{c_1, \dots, c_{15}\}$ *be the set of movies that can be purchased by the airline and* $V = \{v_1, \dots, v_{50}\}$ *be the set of passengers. Each passenger* $v_i \in V$ *submits her favorite subset of movies* $B_i \subseteq C$*. Finally, a voting rule* F *outputs a (set of) recommended movie selection(s)* $F((C, V), 3) \subseteq \mathcal{P}_k(C)$*, which should be available during the flight.*

[20]For further reading on committee elections with approval-based preferences, we refer to the article by Faliszewski, Skowron, Slinko, and Talmon [81] and the textbooks by Brams and Fishburn [40], Laslier and Sanver [120], and Lackner and Skowron [111].

[21]As a convention to avoid improperly defined rules, we always assume that $|C| = m \geq k$ holds.

Next, let us define a large class of voting rules, called *Thiele methods*, introduced by Thiele [167] in 1895. Thiele methods are based on the idea of quantifying the satisfaction of a voter $v_i \in V$ with a possible outcome $W \in \mathcal{P}_k(C)$, by mapping the size of its intersection with her approval ballot $|B_i \cap W|$ to a non-decreasing value. Following the notation by Lackner and Skowron [111], for a non-decreasing function $w\colon \mathbb{N}_0 \to \mathbb{R}$, we define a *scoring function* parameterized by w as $\mathrm{score}_w\colon 2^C \times 2^C \to \mathbb{R}$ with $\mathrm{score}_w(B_i, W) = w(|B_i \cap W|)$ for an individual ballot $B_i \subseteq C$ and a committee $W \in \mathcal{P}_k(C)$. A popular approach for a voting rule F to derive a set of outcomes from the individual scores, is to maximize the *utilitarian social welfare* by selecting those k-committees that maximize the sum of all voters' satisfaction.

Let us portray three prominent examples for utilitarian Thiele methods (also appearing in his original paper [167]): The *(standard) multiwinner Approval Voting rule* (AV), where w_{AV} is the (possibly scaled) identity function; the *Chamberlin-Courant rule* for approval-ballots (CC), named after an ordinal ballot variant by Chamberlin and Courant [50], where w_{CC} is the unit step function; and the *Proportional Approval Voting rule* (PAV), where w_{PAV} is the harmonic series. To group functions that yield an equivalent outcome and exclude trivial scores (i.e., $w(x) = 0$ for all $x \in \mathbb{N}_+$), we set $w(0) = 0$ and $w(1) = 1$. Hence, the above utilitarian voting rules are modeled by

$$F_w((C,V),k) = \operatorname*{arg\,max}_{W \in \mathcal{P}_k(C)} \sum_{v_i \in V} w(|B_i \cap W|)$$

with

$$w_{\text{AV}}(x) = x; \qquad w_{\text{CC}}(x) = \min(x,1); \qquad w_{\text{PAV}}(x) = \sum_{j \in [x]} \frac{1}{j}.$$

For an illustration of a multiwinner election and the (utilitarian) multiwinner Approval Voting rule, see Figure 3.1 in Chapter 3. Analogously, we can maximize the *egalitarian social welfare*, where the overall voters' satisfaction is measured by their worst-off member.[22] Formally, for any scoring function $\mathrm{score}\colon 2^C \times 2^C \to \mathbb{R}$, let the respective egalitarian rule be

$$F((C,V),k) = \operatorname*{arg\,max}_{W \in \mathcal{P}_k(C)} \min_{v_i \in V} \mathrm{score}(B_i, W).$$

Note that mostly throughout literature, a candidate not appearing in an approval ballot is treated as an abstention, rather than a rejection. Hence, a voter is not affected negatively from adding a candidate she does not approve to a committee. As a prominent example for an exception, treating dichotomous preferences as approvals and rejections, the *minimax procedure* by Brams, Kilgour, and Sanver [41] is an egalitarian rule, where a voters' satisfaction with a committee is based on the hamming distance. More precisely, both approved candidates that are not present in a committee and rejected candidates that appear in a committee, are associated with a unit score that should be minimized.

[22] Arguably, egalitarian rules should only be used if voters support a minimum number of candidates. Considering profiles containing empty ballots, the egalitarian social welfare is always zero for any outcome.

Relating back to our initial Example E2, a good choice for a voting rule might be the utilitarian Chamberlin-Courant rule, which maximizes the number of passengers that enjoy at least one movie throughout the flight. In case this number is equal to the number of passengers, the egalitarian Approval Voting rule may maximize the number of movies, the least satisfied voter approves of. Finally, in case the airline changes its policy and charges their customers per stream, the utilitarian Approval Voting rule might maximize the potential revenue. If passengers are less likely to watch multiple movies (or streaming more movies gets increasingly discounted), the Proportional Approval Voting rule might select a more realistic outcome to maximize the companies earnings.

2.2.2 Generalizing Approval Ballots

For some applications, approval ballots are not sufficient to capture the voters' preferences reasonably. Yet, to reduce cognitive burden on the voters, it is preferable to keep the simplicity coming from approval votes (e.g., in contrast to ordinal or cardinal preferences). To give an example, if voters want to express trichotomous preferences (i.e., approval, rejection, or abstention of a candidate), we must move to a slightly generalized ballot format, as discussed by Brams and Fishburn [40] and studied by Baumeister et al. [26, 27]. An extension, where candidates are split into any fixed number of disjoint sets was proposed by Baumeister, Böhnlein, Rey, Schaudt, and Selker [15]. Other examples include ranking only the set of approved candidates [42], conditional approvals modeled by a graph [10], or using expressive languages to logically interconnect atomic approvals [153, 100, 39].

In Chapters 9 and 10, we explore such slight generalizations of approval ballots, which still rely on approvals. Let us continue to introduce respective ballot formats, namely *attribute approval ballots* and *bounded approval ballots*.

Attribute Approval Ballots

Let us investigate situations, where voters care less about the actual candidates themselves, but rather about attributes across different domains either a candidate or an elected committee should satisfy. For example, if we want to elect a committee that is capable of accomplishing a set of tasks, we are less focused on *who* should be elected, as long as the skills expected for success are present. For a systematic study we require a ballot format, allowing for voters to vote on attributes, which in turn may be satisfied by single candidates. An appropriate model has been formally introduced by Kagita, Pujari, Padmanabhan, Aziz, and Kumar [102] under the name *attribute approval elections*,[23] which is explored more detailed in Chapter 9.

[23] Candidates being associated with attributes has also been explored in closely related ways. In *multiwinner elections with diversity constraints* [44], voters vote on candidates directly but any elected committee must abide a mandatory diversification of attributes. Contrarily, in *multi-attribute committee selection* [117], voters specify their target distribution (i.e., the desired diversification of attributes) as ballot format.

Following Kagita, Pujari, Padmanabhan, Aziz, and Kumar [102], in attribute approval elections, we remain in the setting of committee elections, where the task is to elect a set of suitable fixed-size committees; and we still consider a set of candidates C and a set of voters V. Instead of voting on candidates directly, we assume each candidate satisfies a distinct attribute across $d \in \mathbb{N}_+$ non-overlapping categories. Formally, for $j \in [d]$ let D^j be an *attribute domain* with $|D^j| \geq 2$ to exclude trivial domains, and $D^j \cap D^h = \emptyset$ for all $j \neq h$. Then $D = D^1 \times \ldots \times D^d$ specifies the set of all attribute vectors and each candidate $c_i \in C$ is associated with exactly one attribute for each category. This is modeled by a function $a \colon C \to D$, which maps a candidate c_i to her attribute vector $a(c_i) = (c_i^1, \ldots, c_i^d) \in D$. Finally, each voter $v_i \in V$ casts her attribute approval ballot $b_i = (B_i^1, \ldots, B_i^d) \in \mathcal{D} = 2^{D^1} \times \ldots \times 2^{D^d}$, by specifying which subset of attributes $B_i^j \subseteq D^j$ in each category D^j with $j \in [d]$ is thought to be desirable. An attribute approval election E is determined by a tuple $E = (D, C, V)$ and $\mathcal{E}$ holds all such elections. Finally, a voting rule for attribute approval elections F maps an attribute approval election $E \in \mathcal{E}$ along with a positive integer $k \in \mathbb{N}_+$ to a set of winning k-committees $F(E, k) \subseteq \mathcal{P}_k(C)$.

Extending popular voting rules for candidate approval ballots to the attribute approval setting, we study *individual scoring functions* $f \colon \mathcal{D} \times 2^C \to \mathbb{Q}_{\geq 0}$, which map a voter v_i's attribute approval ballot b_i along with a given committee $W \subseteq C$ to a non-negative rational number $f(b_i, W) \in \mathbb{Q}_{\geq 0}$. To give three natural examples, we generalize approval scores and the Chamberlin-Courant scores to this setting, along with a scoring function that explicitly relies on the composition of attributes in a given committee. In particular, disregarding a normalization factor of $1/d$, *Simple Scoring* (f^{si}) maps to the (sum of) approved attributes for a voter's ballot and a committee, *Chamberlin-Courant Scoring* (f^{cc}) maps to the number of satisfied attributes by the most appealing candidate in a given committee, and *Committee Scoring* (f_Σ^{co}) maps to the number of attributes that are satisfied by the group as a collective. More formally, as defined in [21], that is:

$$\textbf{Simple Scoring:} \quad f^{\mathrm{si}}(b_i, W) = \frac{1}{d} \sum_{c \in W} \sum_{j \in [d]} |\{c^j\} \cap B_i^j|$$

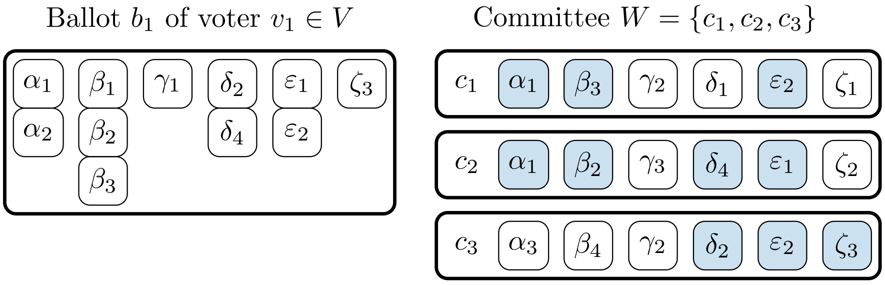
$$\textbf{Chamberlin-Courant Scoring:} \quad f^{\mathrm{cc}}(b_i, W) = \frac{1}{d} \max_{c \in W} \sum_{j \in [d]} |\{c^j\} \cap B_i^j|$$

$$\textbf{Committee Scoring:} \quad f^{\mathrm{co}}(b_i, W) = \frac{1}{d} \left|\left\{j \in [d] \mid \exists c \in W \text{ with } c^j \in B_i^j\right\}\right|$$

Let us illustrate the above individual scoring functions with a simple formal example.

Example E3. *Consider an attribute approval election $E = (D, C, V)$ with $d = 6$ categories, each containing four attributes ($D_1 = \{\alpha_1, \alpha_2, \alpha_3, \alpha_4\}, D_2 = \{\beta_1, \beta_2, \beta_3, \beta_4\}$, and so on). Depicted below are voter v_1's ballot b_1, as well as the attribute vectors for each candidate of a given committee $W = \{c_1, c_2, c_2\}$.[24] Each candidate's attribute that also appears in v_1's ballot is highlighted to improve readability.*

[24]Relating to Example E2, the candidates might be movies, while the attribute categories might be *genre*, *playtime*, *decade of origin*, *leading role*, *director*, and *rating*. Then voters are more focused towards the attributes of a movie, which can be especially relevant if the movies to choose from are unknown in advance.



It is easy to verify, that $f^{si}(b_1, W) = 10/6$*,* $f^{cc}(b_1, W) = 4/6$*, and* $f^{co}(b_1, W) = 5/6$.

For modeling the overall voters' satisfaction, we may extend individual scoring functions in an utilitarian approach (based on summation) or egalitarian approach (based on minimization). For a generic individual scoring function f^x (where $x \in \{\mathrm{si}, \mathrm{cc}, \mathrm{co}\}$), this leaves us with the following *extended scoring functions* $f^x_y \in \{f^x_\Sigma, f^x_{\min}\}$ with

$$f^x_\Sigma(V, W) = \sum_{v_i \in V} f^x(b_i, W) \qquad \text{and} \qquad f^x_{\min}(V, W) = \min_{v_i \in V} f^x(b_i, W),$$

which are maximized by a voting rule $F^x_y((D, C, V), k) = \underset{W \in \mathcal{P}_k(C)}{\arg\max}\, f^x_y(V, W)$.

Bounded Approval Ballots

Next, we consider situations, where voters might want to express a more complex preference, which cannot be captured entirely by a standard approval ballot. As a simple example, illustrated in Figure 2.3, let residents of a multi-party building decide on where to plant a fixed number of trees around the house. Although a person living at a corner apartment may derive (some kind of) satisfaction from trees near her windows, she may be indifferent about plants out of sight. In case of approval ballots she is only able to state which positions she likes. Yet, she is not able to specify that digging up the lawn upfront of her windows is only worthwhile for at least two trees, while there is no additional satisfaction from more than five trees, and more than eight is just too much.

To capture the aforementioned traits — namely dependencies (at least two), substitution effects (five is enough) and incompatibilities (more than eight is too much) — we [23] introduce a novel ballot format as a natural generalization to approval ballots (explored in Chapter 10). A *bounded approval ballot* consists of multiple so-called *bounded approval sets*, each specifying a subset of approved candidates, a lower bound to model dependencies, a saturation point to model substitution effects, and an upper bound to model incompatibilities. As we will be using a slightly different notation for the basic components of committee elections, let us briefly redefine relevant parts. Formally, the set of alternatives is denoted by $\mathcal{A} = \{a_1, \dots, a_m\}$ and the set of voters by $\mathcal{N} = [n]$, each casting a ballot $\boldsymbol{B}_i$, whose format and interpretation will be described in the upcoming paragraph. As per usual, the task is to elect a set of suitable k-committees, i.e., a non-empty subset of $\mathcal{C}_k = \{\pi \subseteq \mathcal{A} \mid |\pi| = k\}$.

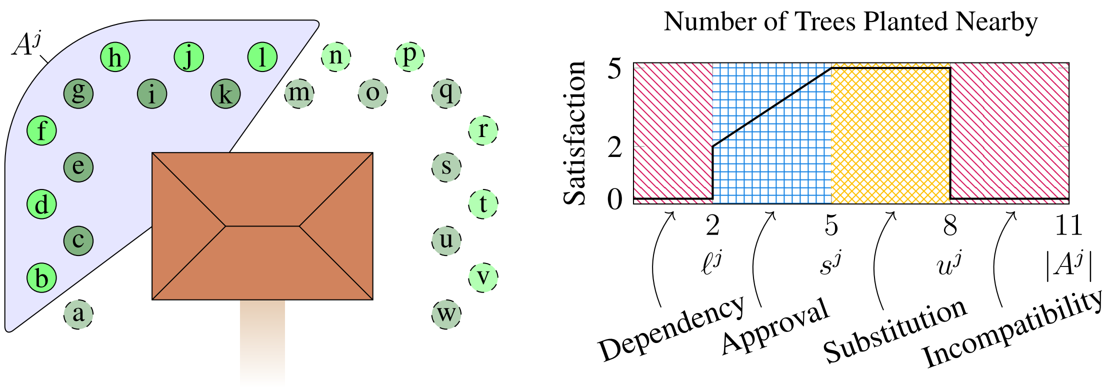


Figure 2.3. Satisfaction derived from the bounded set $B^j = \langle A^j, 2, 4, 8\rangle$ and a potential outcome π (i.e., the set of planted trees), based on the size of the intersection $|A^j \cap \pi|$. The set $A^j = \{b, \dots, l\}$ contains those positions, that are near the (northwestern) corner apartment. If the number of planted trees in A^j falls below the lower bound $\ell^j = 2$ or the above the upper bound $u^j = 8$, no satisfaction is perceived. Otherwise, the satisfaction is increasing steadily with every planted tree until the saturation point $s^j = 5$ is met.

Formally, a bounded approval ballot $\boldsymbol{B} = (B^1, \dots, B^p)$ is a finite list of bounded approval sets. Each such bounded approval set $B^j = \langle A^j, \ell^j, s^j, u^j\rangle$ specifies a non-empty set of approved alternatives $A^j \subseteq \mathcal{A}$ and three integer bounds, modeling the lower bound ℓ^j, the saturation point s^j, and the upper bound u^j, with $1 \leq \ell^j \leq s^j \leq u^j \leq |A^j|$. For a single bounded approval set B^j and a potential outcome $\pi \subseteq \mathcal{A}$, it is rather easy to find a scoring function that behaves as requested. To do so, the relationship of approved alternatives in the committee $|A^j \cap \pi|$ to the given numerical bounds should determine the perceived satisfaction: If $|A^j \cap \pi| < \ell^j$ a dependency is not met and the score should evaluate to zero. Similarly, if $|A^j \cap \pi| > u^j$, the score should be zero due to incompatibility of alternatives. If $\ell^j \leq |A^j \cap \pi| \leq s^j$, all alternatives in $A^j \cap \pi$ are fully approved, resulting in a score of $|A^j \cap \pi|$. If more than s^j (but not more than u^j) alternatives are in the intersection with π, the satisfaction is capped at s^j, as any additional candidates are seen as substitutes.

To reasonably extend those scores from bounded approval sets to full ballots, the score for a bounded set is split equally on all contributing candidates. More precisely, for a given committee π, each alternative $a \in A^j \cap \pi$ contributes to the overall score of $x \in \{|A^j \cap \pi|, s^j, 0\}$ by $x/|A^j \cap \pi| \in \{1, s^j/|A^j \cap \pi|, 0\}$. Formally, an alternative's score under a given bounded set B^j and a committee π can be modeled by the following function φ.

$$\varphi(B^j, \pi) = \begin{cases} 1 & \text{if } \ell^j \leq |A^j \cap \pi| \leq s^j \\ \frac{s^j}{|A^j \cap \pi|} & \text{if } s^j < |A^j \cap \pi| \leq u^j \\ 0 & \text{otherwise.} \end{cases}$$

Finally, we consider four operations for assigning a joint score with respect to the full ballot to every single candidate appearing in a committee. That is, we assign either the minimum, maximum, average, or total score across all contained bounded sets. Formally, when $\boldsymbol{B}_{|a} = \{B^j \in \boldsymbol{B} \mid a \in A^j\}$ refers to those bounded sets in $\boldsymbol{B}$ involving a, the resulting scoring functions are as follows:

$$
\begin{aligned}
score_{min}(\boldsymbol{B}, \pi) &= \sum_{a \in \pi} \min\left\{\varphi(B^j, \pi) \mid B^j \in \boldsymbol{B}_{|a}\right\} \\
score_{max}(\boldsymbol{B}, \pi) &= \sum_{a \in \pi} \max\left\{\varphi(B^j, \pi) \mid B^j \in \boldsymbol{B}_{|a}\right\} \\
score_{avg}(\boldsymbol{B}, \pi) &= \sum_{a \in \pi} \frac{1}{|\boldsymbol{B}_{|a}|} \sum_{B^j \in \boldsymbol{B}_{|a}} \varphi(B^j, \pi) \\
score_{tot}(\boldsymbol{B}, \pi) &= \sum_{a \in \pi} \sum_{B^j \in \boldsymbol{B}_{|a}} \varphi(B^j, \pi)
\end{aligned}
$$

Again, a suitable voting rule should select exactly those k-committees which maximize an underlying scoring function (e.g., one of the scoring functions presented above).

2.3 Participatory Budgeting

Participatory budgeting has gained a decent amount of attention over the course of the last decades. As a participative democratic process, residents of a municipality are involved in deciding on what to spend public funds on. This idea was initially developed and implemented in the 1980s by the Brazilian Worker's Party in Porto Alegre [161]. Following a prolonged story of success, participatory budgeting has been implemented all around the globe [156, 158, 61, 62, 173].

Technically speaking, participatory budgeting can be seen as a natural generalization of multiwinner elections, where each candidate occupies an individual (but fixed) number of seats (i.e., positions in the committee). Shifting slightly in its use-case, the set of alternatives mostly consists of projects, each associated with a predefined cost, which can be funded by a limited budget of funds. To motivate this generalization, let us revisit Example E2 and assume the airline has to pay a varying licensing fee for each movie. Then, a selection of movies may not be limited by a fixed number of movies, but by a fixed budget of funds which should not be exceeded by the resulting licensing cost.

For a formal overview over this research field through the lens of computational social choice we refer to the book chapter by Aziz and Shah [8]. A yet to publish survey paper by Rey and Maly [150] extensively discusses the current state of the art. Similar to multiwinner elections, there is a densely populated literature on different ballots formats, including cardinal [142, 30, 78] and ordinal preferences [7, 30, 78], while this thesis focuses on approval-based preferences. We define approval-based participatory budgeting in the upcoming subsection formally.

2.3.1 Approval-based Participatory Budgeting

Although there are several similar frameworks for participatory budgeting considering approval ballots [6, 30, 110], we follow the formal model and notation by Talmon and Faliszewski [166] in its irresolute variant [25].

Formally, a participatory budgeting campaign (also called *budgeting scenario*) is a tuple $E = (A, V, c, \ell)$, consisting of the following four basic components. As analogue components to multiwinner elections, we consider a set of m alternatives (called *projects* or *items*) $A = \{a_1, \dots, a_m\}$ and a set of n voters $V = \{v_1, \dots, v_n\}$. Each voter $v \in V$ submits her approval ballot by specifying the subset of projects $A_v \subseteq A$ she would like to see being implemented. Deviating from committee elections, each alternative $a \in A$ is associated with a positive integer cost $c(a)$, modeled by a *cost function* $c\colon A \to \mathbb{N}_+$.[25] Lastly, a *budget limit* $\ell \in \mathbb{N}_+$ dictates the overall available funds, which should not be

[25]To ease notation, for a set of projects $B \subseteq A$, we write $c(B) = \sum_{a \in B} c(a)$.

exceeded by any implemented *bundle* $B \subseteq A$.[26] In particular, a given bundle $B \subseteq A$ abides *feasibility* if and only if $c(B) \leq \ell$ holds. Some authors specify further reasonable restrictions. For example, Benadè, Nath, Procaccia, and Shah [30] assume that a valid ballot must also abide the given budget constraint (also called *knapsack vote* [30, 28]), while often we wish for the outcome to be *exhaustive* [6, 166, 148], i.e., not extendable without violating feasibility. Lastly, a common assumption is that a single project may never exceed the budget limit, effectively restricting the cost function to $c : A \to [\ell]$.

To quantify the agreement of a voter with a possible outcome (i.e., a feasible bundle), Talmon and Faliszewski [166] conceptually extend the idea by Thiele [167]. In particular, a *satisfaction function* $s\colon 2^A \times 2^A \to \mathbb{N}_0$ models the individual satisfaction perceived by a voter $v \in V$ with ballot $A_v \subseteq A$ for a given bundle $B \subseteq A$. As the considered functions rely on the intersection of a voter's ballot and a bundle, let this be denoted by $B_v = A_v \cap B$ as a useful notation. The authors adopt approval scores and Chamberlin-Courant scores from multiwinner elections to the budgeting setting, to measure the satisfaction of a voter by the *quantity* ($|B_v|$) or *presence* ($\mathbb{1}_{|B_v|>0}$) of projects she approves of. Additionally, satisfaction by *cost* ($c(B_v)$) models that a voter's happiness correlates with the amount of funds spent on projects she likes. Formally, those functions can be modeled as follows:

$$\begin{aligned}
&\textbf{Quantity:} && s(A_v, B) = |B_v| && = w_{\text{AV}}(|B_v|)\\
&\textbf{Presence:} && s(A_v, B) = \mathbb{1}_{|B_v|>0} && = w_{\text{CC}}(|B_v|)\\
&\textbf{Cost:} && s(A_v, B) = c(B_v) &&
\end{aligned}$$

To derive a collective outcome based on the voters' satisfaction, we may use similar operators as for multiwinner elections. In particular, Talmon and Faliszewski [166] mostly focus on utilitarian optimization, by either maximizing the (sum of) voters' satisfaction or approximate the optimal result by greedily adding either the project with the maximum additional satisfaction or the project with the best satisfaction-to-cost ratio.[27] We denote those composite rules (consisting of a satisfaction function and an aggregator) by *max rules* ($\mathcal{R}_s^m$), *greedy rules* ($\mathcal{R}_s^g$) and *proportional greedy rules* ($\mathcal{R}_s^p$). Note that we often interpret all those rules as irresolute [25], assuming parallel-universe tie-breaking for (proportional) greedy rules. Additionally, we combine both greedy approaches (by pre-computing one bundle each and outputting the better one) to study *hybrid greedy rules* ($\mathcal{R}_s^h$) which yield a constant approximation factor [82].

We want to introduce one additional rule, which is becoming increasingly popular due to its proportional distribution of funds. The *Method of Equal Shares* (formerly known as *Rule X*) originated from a multiwinner variant by Peters and Skowron [144] and was generalized by Peters, Pierczyński, and Skowron [142] to the setting of participatory budgeting. It is a sequential rule which simulates how voters might fund projects on their own,

[26] To avoid confusion, it is worth noting that due to a slight inconsistency throughout literature, the term *budget* refers to the budget limit ℓ in some articles, while to a bundle $B \subseteq A$ in others.

[27] In related approaches, Sreedurga, Bhardwaj, and Narahari [162] use an egalitarian operator and Fluschnik, Skowron, Triphaus, and Wilker [83] aim to maximize the Nash social welfare.

if every participant was given her proportional share of the budget in advance. Informally, the Method of Equal Shares aggregates as follows. Initially, the budget limit ℓ is spread equally across the n voters, each receiving a deposit of ℓ/n. Then, each round exactly one project is funded by its supporters (spending their individual deposits), until there is no affordable project left (assuming voters do not fund any project they do not approve of). Just as simple it is to decide, which project will be funded in any round. For every project it is calculated how much each supporter has to pay, if the cost is distributed equally. Those supporters that cannot afford to pay their equal share give all of their remaining budget, while the difference must be paid by the remaining supporters. Finally, the project with the lowest equal share (i.e., the amount the worst-off supporter has to pay) is selected, implemented and paid for by its supporters (using a tie-breaking scheme if necessary).

2.3.2 Uncertain Project Costs

In this subsection, we formally extend the basic framework for approval-based participatory budgeting, by considering uncertainty on the implementation cost of any project. This idea was first studied in a broad stochastic model by Gomez, Insua, and Alfaro [87], where uncertainty is posed on all parameters in a budgeting campaign. Closely related, our framework [24] (explored in Chapter 6) is motivated by real-world applications, where particularly the exact cost of a project is only revealed after its implementation. Therefore, we assume the cost of a project is rather given as an estimate, coming from a probability distribution. Furthermore, each project is associated with an implementation duration while a given time frame (e.g., a legislation period) should not be exceeded. By considering implementation durations after which the exact cost of a project is revealed, an algorithm can act in an *online* fashion and decide to implement a project after more information (i.e., some other project's exact cost) becomes available [82]. We illustrate this in the following example.

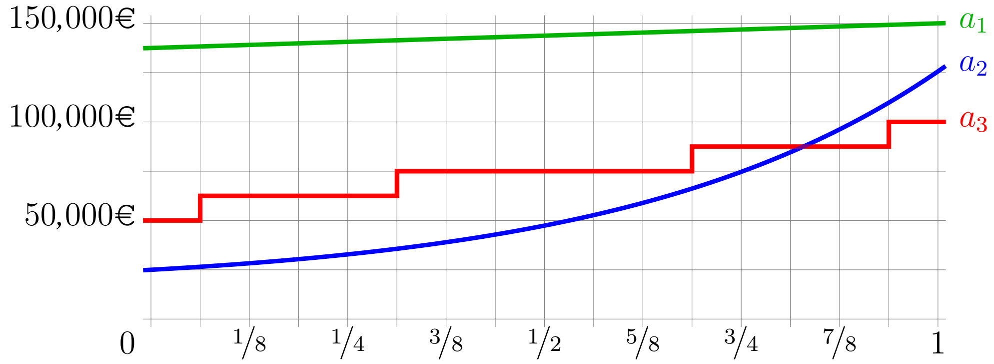
Example E4. *Consider a participatory budgeting campaign, where citizens decide, which of the following projects should be implemented over the next five years.*

Project $\mathbf{a_1}$: *Bike lanes can be added to all major roads. While the cost for this can be estimated quite precisely, its implementation is expensive and takes five years due to legal reasons. Project a_1 was approved by* 200 *voters.*

Project $\mathbf{a_2}$: *Solar panels can be installed on some buildings of the local university. The installation can be done in one year, but due to supply shortages, it is unclear whether the overall cost might explode. Project a_2 was approved by* 300 *voters.*

Project $\mathbf{a_3}$: *The library, which was eventually closed due to structural damages, can be restored. This process would take three years, but for each of the four main walls there is a fifty percent chance that it has to be replaced for an additional cost. Project a_3 was approved by* 400 *voters.*

To exemplify the given projects' costs, the following diagram models suitable probability distributions for each of the three projects. A point (x, y) *on the curve for project* a_i *marks the probability* $x \in [0, 1]$*, that the exact cost for* a_i *is at most* y*.*



Assuming a budget limit of 325,000€*, there are multiple reasonable strategies. For example, we could implement* a_2 *and* a_3*, as those two projects receive a total of* 700 *approvals. Yet, even if it turns out after one year, that the cost for* a_2 *is at most* 75,000€ *(meaning all three projects are affordable), we cannot begin to implement* a_1 *without exceeding the given time frame of five years. In a different approach, we could implement* a_1 *and* a_3 *first, which receive* 600 *approvals. If after three years it turns out that* a_3 *only costs* 50,000€*, we can also safely implement* a_2*. Note that this approach has only a* $^1/_{16}$ *chance of implementing all three projects (i.e., if no wall of the library has to be replaced). Therefore, a third approach may begin by implementing* a_1 *and* a_2*. Although those two projects only receive* 500 *approvals, chances are about* $^3/_4$ *that* a_2 *will be cheap enough to safely start the implementation of* a_3 *after one year.*

Formally, a budgeting scenario with uncertain cost is given by a tuple $E = (A, V, \widetilde{c}, \delta, \tau)$. In this setting, the parameters A, V, and ℓ again model the set of alternatives, the set of voters and the budget limit. The cost function is replaced by a total of four such functions $\widetilde{c} = (c_{\min}, c_{\max}, c, c_p)$, where $c_{\min}(a)$, $c_{\max}(a)$ and $c(a)$ respectively model the *lower bound*, *upper bound*, and *exact cost* of a project $a \in A$. By design it holds that $c(a) \in [c_{\min}(a), c_{\max}(a)]$. To model the probability distribution on the cost for a project $a \in A$, let $c_p(a, y) \in \mathbb{R}_{\geq 0}$ denote the probability that a has a cost of at most $y \in \mathbb{N}_+$. Finally, each project $a \in A$ is equipped with a *duration* $\delta(a) \in \mathbb{N}_+$ and we assume that a given *time limit* $\tau \in \mathbb{N}_+$ should not be exceeded by an implementation process.

To capture an *online budgeting method's* output in a suitable data structure, we use a *budgeting log* to keep track of the starting time of each implemented project. More precisely, a budgeting log $L\colon A \to \mathbb{N}_0 \cup \{\bot\}$ is a simple function, which maps every project to either a discrete starting time or to a distinct symbol ($\bot$), indicating a project has not been implemented at all. Projects that are *realized* are collected by a set $R(L) = \{a \in A \mid L(a) \neq \perp\}$. Furthermore, an online budgeting method $\mathcal{R}$ is given a budgeting scenario E and sequentially builds a budgeting log L as follows. Starting with an empty budgeting log L with $L(a) = \perp$ for all $a \in A$ and an initial time step $t = 0$, $\mathcal{R}$ can always start implementing a new project $a \in A \setminus R(L)$ by setting $L(a) = t$ or progress in time by increasing t (which corresponds to waiting for a certain amount of time). Any online budgeting method $\mathcal{R}$ has limited access to the (exact) cost function c and may access $c(a)$ for a project $a \in R(L)$ only after implementation, i.e., at time step $t \geq L(a) + \delta(a)$. Finally, at any point in time, $\mathcal{R}$ is allowed to terminate and output its current budgeting log L.

2.4 Judgment Aggregation

Judgment aggregation is a powerful framework, capable to compactly model many real-world applications, where judges share their opinion over multiple interconnected binary issues, which should be aggregated into a feasible collective outcome. In contrast to multiwinner elections or participatory budgeting, a judge's opinion is interpreted as a symmetric preference, where not voting for an issue implies a preference for its rejection. At this point, it is worth noting that there are multiple equally expressive frameworks, each having its own perks and drawbacks. Let us begin by formally defining *"standard"* (i.e., formula-based [124]) judgment aggregation, as described by Endriss [71] (following notation conventions by Endriss, Grandi, and Porello [73]) in the upcoming subsection. We introduce a variety of consistent judgment aggregation rules in Subsection 2.4.2 and explore recent extensions and variations of judgment aggregation in Subsections 2.4.3 and 2.4.4.

2.4.1 Formula-based Judgment Aggregation

Let us briefly introduce some notation, to simplify the upcoming definition. We call two propositional formulas φ and ψ *equivalent*, denoted by $\varphi \equiv \psi$, if they admit the same set of satisfying truth assignments $L(\varphi) = L(\psi)$; whereas they are equal, denoted by $\varphi = \psi$ if the formulas are identical. We call a propositional formula φ doubly-negated if $\varphi = \neg\neg\psi$ for a propositional formula ψ; and if φ is not doubly-negated, we denote by $\sim\!\varphi$ the complement of φ, that is, $\sim\!\varphi = \psi$ if $\varphi = \neg\psi$ (for a formula ψ) and $\sim\!\varphi = \neg\varphi$, otherwise. We call a set of propositional formulas $\{\varphi_1, \ldots, \varphi_k\}$ consistent, if there is a truth assignment satisfying all formulas simultaneously, i.e., $\bigcap_{i\in[k]} L(\varphi_i) \neq \emptyset$. Finally, two formulas φ and ψ are independent, if the sets $\{\varphi, \psi\}$, $\{\sim\!\varphi, \psi\}$, $\{\varphi, \sim\!\psi\}$ and $\{\sim\!\varphi, \sim\!\psi\}$ are all consistent.

Definition (Formula-based Judgment Aggregation)**.** *Let $\mathcal{L}$ be the set of all propositional formulas. In judgment aggregation, we are given a non-empty and finite* agenda *$\Phi = \{\varphi_1, \neg\varphi_1, \ldots, \varphi_m, \neg\varphi_m\} \subset \mathcal{L}$, which consists of $m \in \mathbb{N}$ issues (that is, a formula and its complement) and must abide some basic requirements.*

The agenda Φ does not contain any doubly-negated formulas, is closed under complement,[28] nontrivial (i.e., contains at least two independent issues), and does not contain any contradiction ($\varphi \equiv 0$).[29] A judgment *$J \subseteq \Phi$ is called* complete *if $|\{\varphi, \sim\!\varphi\} \cap J| \geq 1$ for all $\varphi \in \Phi$ and* complement-free *if $|\{\varphi, \sim\!\varphi\} \cap J| \leq 1$ for all $\varphi \in \Phi$. By $\mathcal{J}(\Phi) \subset 2^{\Phi}$ we denote the set of all complete and consistent judgments. A set $\mathcal{N} = [n]$ of $n \in \mathbb{N}$* judges *takes part in a collective decision over the given agenda. There-*

[28]Sometimes written as $\Phi = \Phi^+ \cup \Phi^-$ with $\Phi^- = \{\sim\!\varphi \mid \varphi \in \Phi^+\}$ and $\Phi^+ \cap \Phi^- = \emptyset$, effectively splitting the considered issues into a *positive* and *negative* agenda.

[29]Note that by closure under complement, Φ may also not contain a tautology ($\varphi \equiv 1$). Some frameworks relax the formal requirements by allowing for contradictions or trivial agendas.

fore, each judge $i \in \mathcal{N}$ provides an individual judgment $J_i \in \mathcal{J}(\Phi)$, *forming a profile* $P = (J_1, \ldots, J_n) \in \mathcal{J}(\Phi)^n$.

Finally an irresolute judgment aggregation rule $F\colon \mathcal{J}(\Phi)^n \to 2^{2^\Phi}$ *maps from a profile of individual judgments to a collective outcome, which is a set of (aggregated) judgments. If for any agenda* $\Phi \subset \mathcal{L}$ *and any profile* $P \in \mathcal{J}(\Phi)^n$*, all judgments in* $F(P)$ *satisfy a given property (e.g., completeness, complement-freeness, consistency), we say that* F *satisfies this property, too. As a special case for irresolute rules, we define resolute rules that output exactly one judgment as outcome by* $R\colon \mathcal{J}(\Phi)^n \to 2^\Phi$.

As the arguably most intuitive example for a reasonable judgment aggregation rule, consider the (strict) *majority rule*, which selects a single outcome based on the majority-wise decision for each issue. Formally, for any agenda Φ and profile $P \in \mathcal{J}(\Phi)^n$, the majority rule Maj is a resolute rule and defined as

$$\text{Maj}(P) = \{\varphi \in \Phi \mid |\{i \in \mathcal{N} \mid \varphi \in J_i\}| > {}^n\!/\!_2\}.$$

It is easy to see, that the majority rule is not complete, as a profile with an even number of judges does not necessarily contain a strict majority for each issue. In fact, modeling a (reasonable) complete and consistent judgment aggregation rule is not a trivial task, as the majority-wise decision for each issue may also result in inconsistency. This phenomenon has been initially presented by Kornhauser and Sager [107] in form of the famous *doctrinal paradox*, which was formally revisited as the *discursive dilemma* by List and Pettit [124] as follows.

Example E5. *Three judges should decide, whether a defendant is liable for breaching a contract. By law, the defendant is liable, if both the contract is legally binding and the defendant did indeed violate the contract's agreement. While the first judge comes to the conclusion, there was no breach under a valid contract; the second judge observes an invalid contract, which would have been breached; and the third judge thinks that the defendant is liable by breaching a valid contract. Formally, we can model this scenario by the following judgment aggregation setting: Let* $\Phi^+ = \{c, b, c \wedge b\}$*, where the premises* c *and* b *respectively represent a valid contract and a breach, while its conclusion* $c \wedge b$ *models whether the defendant is liable. Then, the profile of individual judgments* $P = (J_1, J_2, J_3)$ *is given by the following table:*

	c	b	$c \wedge b$
J_1	1	0	0
J_2	0	1	0
J_3	1	1	1
Maj(P)	1	1	0

As we see, although every individual judgment $J_i \in P$ *is complete and consistent, the outcome of the majority rule Maj, i.e., the set Maj*$(P) = \{c, b, \neg(c \wedge b)\}$*, is inconsistent.*

Paradoxically, the defendant might be both, guilty and innocent, depending on whether a majority is posed on either the premises or the conclusion. Overall, under majority-wise aggregation per issue, the defendant must have breached a valid contract without being liable.

At least completeness and complement-freeness can be achieved by simple, issue-wise aggregation rules. A large class of such rules, introduced by Dietrich and List [64], are so-called *quota rules*, which generalize the majority rule by defining a quota on the support for each element of the agenda. The authors show, in case the quotas for an element and its complement add up to $n+1$ (where n is the number of judges), the resulting quota rule is complete and complement-free.

Note that, although inconsistent, using quota rules and, in particular, the majority rule does not only appear to be rather intuitive, but the respective outcomes satisfy a variety of desirable axiomatic properties. In fact, we may easily find a complete and consistent judgment aggregation rule by disregarding basic requirements any reasonable judgment aggregation rule should satisfy. For example, dropping the property of anonymity (i.e., all judges should be treated equally) paves the way for *dictatorship*, a simple rule where a fixed judge may decide the outcome by herself.

2.4.2 Consistent Judgment Aggregation Rules

To overcome the discursive dilemma, there are countless ways to design judgment aggregation rules that are complete and consistent (see List and Puppe [126]). Yet, apart from rare exceptions, calculating the aggregated result for complete and consistent judgment aggregation rules is computationally hard, including all rules that are defined in the remainder of this chapter.[30] One of the most popular rules, namely the *median rule*, studied by Nehring, Pivato, and Puppe [134],[31] selects exactly those complete and consistent judgments that minimize the hamming distance over the individual judgments in a profile. In an egalitarian fashion, the *egalitarian median rule*[32] selects all judgments that minimize the hamming distance for the worst-off judge.

A large stream of research studies majority-preserving rules, where, in case of completeness and consistency, the outcome of the majority rule is selected. In Chapter 7, we generalize majority-preserving rules by considering arbitrary resolute, complete, and complement-free underlying rules, whose outcome may be preserved. Formulated in this generalized variant, we study *sequential rules* [13] (as suggested by List [125]), where the acceptance of issues is decided in a predefined order.[33] By default, the decision for

[30] For an overview of such complexity results we refer to Endriss, de Haan, Lang, and Slavkovik [75].

[31] The median rule has been studied under several aliases: *Distance-based procedure* [145, 73], *prototype rule* [131], *maximum-weight subagenda rule*[115, 119], *max-sum rule* [76], and *Kemeny rule* [74], generalizing the famous rule by Kemeny [105].

[32] Also studied under the names d_H*-max rule* [115] and *MaxHam(ing) rule* [92, 75, 36].

[33] Resulting rules are also studied under the name *sequential priority procedure* [126].

an issue by an underlying rule is preserved, deviating only to guarantee consistency. For the majority rule as underlying rule, the resulting aggregation method has been studied by Peleg and Zamir [141] as *sequential majority rule*. If additionally, the processing order is based on the issue-wise support by the judges, we consider the *ranked agenda rule* by Lang, Pigozzi, Slavkovik, and Torre [115],[34] where we consider either a fixed tie-breaking for a resolute variant [74] or parallel universe tie-breaking for an irresolute variant [74, 119, 118]. In a non-sequential, majority-preserving approach, the *maximum subagenda rule*,[35] studied by Lang, Pigozzi, Slavkovik, and Torre [115], as well as Lang and Slavkovik [118], selects exactly those judgments, which contain an inclusion maximal subset (with respect to consistency) of the majority outcome (or some other underlying rule in a generalized variant). As a refinement, Lang, Pigozzi, Slavkovik, and Torre [115] studied the *maxcard subagenda rule*,[36] which selects exactly those complete and consistent judgments, whose intersection with the majority outcome is maximized.

2.4.3 Judgment Aggregation Extensions

We continue to briefly introduce equally expressive judgment aggregation frameworks and recent extensions. To actually compare the *expressiveness* across frameworks, note that by a slight relaxation of the standard, formula-based framework, we are able to model the set of complete and consistent judgments $\mathcal{J}(\Phi)$ at will (by choosing the agenda Φ accordingly). More precisely, by allowing for contradictions and trivial agendas, Dokow and Holzman [69] showed for a universal agenda $\Phi = \{\varphi_1, \neg\varphi_1, \dots, \varphi_m, \neg\varphi_m\}$ (where each formula initially only acts as placeholder) and any set of complete and complement-free judgments $X \subset 2^\Phi$ (i.e., with $|\{\varphi_i, {\sim}\varphi_i\} \cap J| = 1$ for all $J \in X$ and all $\varphi_i \in \Phi$), how to construct the formulas $\varphi_i \in \mathcal{L}$, such that $\mathcal{J}(\Phi) = X$ holds. On a more abstract level and with a slight abuse of notation, we may either identify each propositional formula by a binary value to represent any judgment J by a binary string of size $2m$, or identify each issue (i.e., a formula and its complement) by a binary value to represent any complete and complement-free judgment by a binary string of size m. Hence, following the results of Dokow and Holzman [69], for any fixed-size binary space $X \subseteq \{0,1\}^m$, there is an agenda Φ with $\mathcal{J}(\Phi) = X$.

Grandi [89] and Endriss [88] proposed an equally expressive judgment aggregation framework, known as *binary aggregation with integrity constraints*, where the agenda consists of propositional variables, interconnected by an external propositional formula (called integrity constraint) any judgment must abide.[37] Note that, in formula-based judgment aggregation, it is NP-hard to verify whether a complete judgment is consistent (by corre-

[34] This rule is also known as *Tideman (ranked pairs) rule* [74] or *support-based procedure* [146]. Its refinement based on the lexicographical order of the issues is known as *leximax rule* [132, 77].

[35] Also called *Condorcet admissible set* [134], *(maximal) Condorcet rule* [116, 75], or *max-set rule* [76].

[36] Called *endpoint rule* [131], *Slater rule* [74, 75], *maxcard Condorcet rule* [116], and *max-num rule* [76].

[37] Unrestricted expressiveness is easy to verify, as an external propositional constraint in disjunctive normal form can list every satisfying assignment as a clause.

sponding to a SAT problem), while in binary aggregation, it can be easily done in polynomial time by checking whether a given truth assignment satisfies the external constraint (by sequentially resolving its connectives). Hence, unless $\mathrm{P} = \mathrm{NP}$, an equivalent external constraint might result in exponential length.

Structurally building on the idea that an equivalent framework may encode constraints more compactly, Endriss, Grandi, de Haan, and Lang [72] provided an extensive study on the *succinctness* of different, equally expressive judgment aggregation frameworks. In particular, they considered that the set of complete and consistent judgments is either given as a binary space explicitly, modeled by a formula-based agenda, an integrity constraint with or without additional propositional variables, or a mixture of the last two. They also showed that, for some judgment aggregation rules, there is a computational complexity gap across frameworks which are not equally succinct.

Endriss [76] extended the framework by Grandi [89] and Endriss [88], by introducing *rationality* and *feasibility* constraints to allow for non-identical input and output spaces in judgment aggregation. This simple generalization significantly expands the scope of judgment aggregation, enabling the simulation of ballot formats beyond the scope of basic approval preferences. For example, to model fixed-size committee elections with ordinal ballots, it suffices to simulate a binary relation by creating an agenda item for each (ordered) pair of candidates. Now, a rationality constraint may model strict rankings in the input by requiring completeness, asymmetry, and transitivity (for a formal encoding see Dietrich and List [63]), while feasibility may be imposed by restricting the outcome to be a dichotomous order, partitioning the candidates into two sets of desired sizes. Particularly, Chingoma, Endriss, and de Haan [51] showed how to simulate multiwinner election rules in judgment aggregation (considering both ordinal and approval-based preferences).

2.4.4 Weighted Asymmetric Judgment Aggregation Rules

In contrast to participatory budgeting, a large stream of research considers only those judgment aggregation rules that interpret the issues as binary and the preferences as symmetric. That is, each issue is treated uniformly and the rejection of an issue is treated as an approval for its complement (instead of an abstention). Recently, Nehring and Pivato [133] relaxed the former requirement to study weighted judgment aggregation rules, while Rey, Endriss, and de Haan [149, 148] relaxed the latter requirement to study asymmetric judgment aggregation rules.

A simple way to model weighted and/or asymmetric rules is by interpreting known rules based on the minimization of the hamming distance as scoring rules, where the goal is to maximize a score. Formally, scoring rules have been introduced by Dietrich [65], who also showed that the median rule can be reformulated as a scoring rule, which assigns each issue in the intersection of an individual judgment and a possible outcome a score of one. To study asymmetric judgment aggregation, following Rey, Endriss, and de Haan [149, 148], it is sufficient to restrict the contribution to the overall score to the issues of the positive agenda. In the weighted (but symmetric) variant, studied by Nehring and Pivato [133], each issue is equipped with an (e.g., real valued) weight, while the score of an outcome and an individual judgment is given by the added weight in the intersection.

As technically explored in Chapter 8, we combine both relaxations to model *weighted, asymmetric judgment aggregation* rules [34]. This allows us to encode many collective combinatorial optimization problems (and domain-specific rules) into a general judgment aggregation framework. By allowing for arbitrary constraints (e.g., linear equations), the succinctness of a judgment aggregation framework can be improved.[38]

Formally, deviating slightly in notation from established conventions by Endriss, Grandi, and Porello [73], an agenda A is a finite set of atomic propositions, such that A_+ holds all positive issues and A_- their complements. Additionally, each issue $a \in A$ is associated with an integer weight $w_a \in \mathbb{N}_0$, collected in a weight vector $\boldsymbol{w}$. Rationality constraints Γ_R and feasibility constraints Γ_F (both possibly incorporating the given weights) can be imposed to model the set of valid input judgments $\mathcal{B}_R \subseteq \{0,1\}^{|A|}$ and the feasible outcomes $\mathcal{B}_F \subseteq \{0,1\}^{|A|}$. In this representation, a vector $X \in \{0,1\}^{|A|}$ specifies an assignment over propositional variables, where $X(a) = 1$ if and only if the entry of X at position $a \in A$ is set to one. For a judge $i \in \mathcal{N}$, we represent her individual judgment as a ballot $B_i \in \mathcal{B}_R$. Finally, by restricting the weights, we are able to model scoring rules that can be weighted (w_a can differ from being zero or one for all $a \in A$) and/or asymmetric ($w_{\neg a} = 0$ for all $\neg a \in A_-$).

To illustrate how this generalization affects previously defined voting rules, the (weighted, asymmetric) median rule can then be reformulated as follows:

$$\operatorname*{arg\,max}_{X \mid X \in \mathcal{B}_F} \sum_{i \in \mathcal{N}} \sum_{a_j \in A} w_{a_j} \cdot B_i(a_j) \cdot X(a_j)$$

Note that, in case of uniform weights (i.e., $w_a = 1$ for all $a \in A$), this rule coincides with the standard median rule. Similarly, we can generalize the egalitarian median rule by swapping the first sum operator for a min operator. For sequential rules, including the ranked agenda rule, it is sufficient to base the order of issues on the weighted support. Finally, considering an asymmetric setting, the Chamberlin-Courant rule for approval ballots [159] can be meaningfully translated into a judgment aggregation rule (as for symmetric agendas a judge is only dissatisfied in case the outcome is complementing her individual judgment). We explore this general framework in more depth in Chapter 8.

[38] In particular, Rey, Endriss, and de Haan [148] showed, that encoding a compact budget constraint into propositional logic might require exponential space unless the polynomial-time hierarchy collapses.

MOTIVATION AND RESEARCH QUESTIONS

The main focus of this thesis is to investigate the three closely related research fields of *multiwinner elections* with approval ballots, *participatory budgeting*, and *judgment aggregation*. Our aim is to study the respective research areas individually on a local level, as well as through a more abstract lens on a global level. To study the latter, let us take a step back and investigate a more general perspective on the frameworks, uncovering the shared structure of all three independently formalized models.

Informally, in all three frameworks we are given a finite set of alternatives and a finite number of participants, that individually specify which subset of alternatives are desired in an outcome. Then, a voting (or aggregation) rule is used to determine which subset(s) of alternatives should be in a collective outcome. On an abstract level, deriving a collective decision on multiple binary alternatives from approval preferences can be represented by a simple class of functions. Note that the decision on a fixed number of $m \in \mathbb{N}$ binary issues (both, on an individual level in the input and on a collective level in the output) can be represented by a binary string of length m. Further, there might be restrictions posed for an input string to be valid, or for an output string to be feasible. Formally, let $\mathcal{X} \subseteq \{0,1\}^m$ be a predefined set of valid inputs and $\mathcal{Y} \subseteq \{0,1\}^m$ be a set of feasible outputs. The input then consists of a list of $n \in \mathbb{N}$ valid binary strings $X \in \mathcal{X}^n$ (each representing the opinion of an individual agent) and the output is a set of feasible strings $Y \subseteq \mathcal{Y}$. To model the mapping of an input to an output, we consider a function $F : \mathcal{X}^n \to 2^{\mathcal{Y}}$, which concludes our general setup.[39] Figure 3.1 illustrates a simple multiwinner election in this abstract model, where five voters provide approval ballots over eight alternatives (as a list $X \in \mathcal{X}^5$). The task is to select suitable committees (i.e., a set $Y \subseteq \mathcal{Y}$) consisting of exactly five alternatives, using the Approval Voting rule (F with $F(X) = Y$).

[39] A coinciding framework for binary aggregation has been formally introduced by Dokow and Holzman [69, 68] and studied under the name *BASIC* by Endriss, Grandi, Haan, and Lang [72].

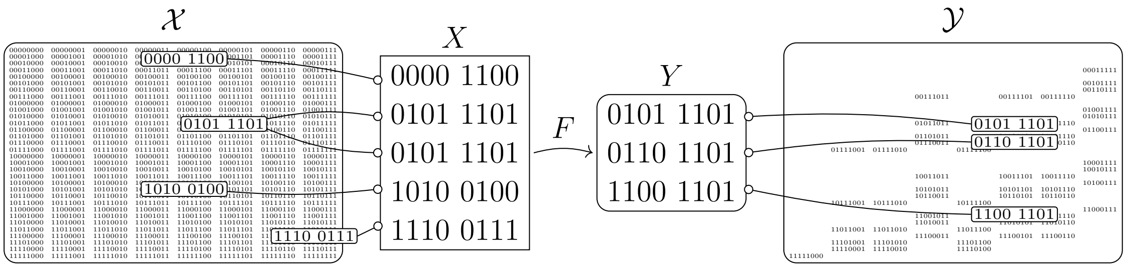


Figure 3.1. Example of a multiwinner election with eight candidates and five voters electing a committee of size $k = 5$. The set of valid inputs is unrestricted, i.e., $\mathcal{X} = \{0,1\}^8$, and the set of feasible outputs is modeled by a fixed-size committee constraint, i.e., $\mathcal{Y} = \{y \in \{0,1\}^8 \mid hw(y) = 5\}$.[40] As voting rule F we consider the standard multiwinner Approval Voting rule (AV), which selects those k-committees that maximize the overall (sum of) approved alternatives.

Of course, there are countless ways to choose the three main components $\mathcal{X}$, $\mathcal{Y}$, and F, in an attempt to model real-world applications realistically. It turns out that the fields of multiwinner elections, participatory budgeting, and judgment aggregation can be solely characterized by choosing $\mathcal{X}$ and $\mathcal{Y}$ accordingly. Let us illustrate how we can model restricted domains for our basic setup to characterize each research field respectively. To do so, we will describe how each pair $(\mathcal{X}, \mathcal{Y})$ must be shaped in order to fall into one of the respective research fields.

Multiwinner Elections

In (approval-based) multiwinner elections, there are no restrictions on how participants may vote on a set of given alternatives $A = \{a_1, a_2, \ldots, a_m\}$. Therefore, the set of valid input strings is given by $\mathcal{X} = \{0,1\}^m$. The goal is to select a committee of candidates of predefined size $k \in \mathbb{N}_+$. In order to compactly model the output space, consider $\mathcal{Y} = \{y \in \{0,1\}^m \mid hw(y) = k\}$.[40]

Participatory Budgeting

In participatory budgeting, each alternative $a_i \in A$ is associated with a positive integer cost $c(a_i) \in \mathbb{N}_+$ and we are given a budget limit $\ell \in \mathbb{N}_+$, that should not be exceeded by the (sum of) alternatives' costs in a feasible outcome. Therefore, the output space can be modeled by a budget constraint using a single inequality, i.e., $\mathcal{Y} = \{y_1 y_2 \ldots y_m \in \{0,1\}^m \mid \sum_{i \in [m]} y_i \cdot c(a_i) \leq \ell\}$. As in multiwinner elections, the input space is usually unconstrained, i.e., $\mathcal{X} = \{0,1\}^m$.

[40] Recall that $hw(y)$ refers to the hamming weight of y, effectively modeling the desired committee size.

Judgment Aggregation

As extensively discussed in Section 2.4, there are multiple slightly different frameworks for judgment aggregation. For the sake of simplicity, we now consider the framework by Grandi and Endriss [88], also known as binary aggregation with integrity constraints. Here, we consider a set of binary propositional variables $P = \{p_1, \ldots, p_m\}$, which are logically interconnected by a propositional formula φ as constraint over the variables in P. Both, a voter's valid ballot and any feasible outcome must abide the given logical constraint, i.e., $\mathcal{X} = \mathcal{Y} = L(\varphi)$. As any subset of $\{0,1\}^m$ can be induced by a suitable logical constraint, it is sufficient to require that $\mathcal{X} = \mathcal{Y} \subseteq \{0,1\}^m$. Considering the framework with rationality and feasibility constraints by Endriss [76], inputs and outputs may be modeled arbitrarily, i.e., $\mathcal{X}, \mathcal{Y} \subseteq \{0,1\}^m$.

Relationships Between Frameworks

Let us briefly discuss relationships across all three frameworks (or recent variations) and their implications. It is rather easy to see, that every multiwinner election corresponds to an exhaustive participatory budgeting campaign [6].[41] In turn, every (possibly exhaustive) participatory budgeting campaign can be encoded into the judgment aggregation framework by Endriss [76] by only restricting the set of outputs (see Rey, Endriss, and de Haan [148]). If we consider the (binary aggregation) framework by Grandi and Endriss [88] (as specialization of the framework by Endriss [76]), it is easy to see that we may encode any participatory budgeting campaign, where ballots must also abide the budget constraint (known as knapsack vote [29]). Observing those frameworks in a generalization-to-specialization relationship allows axiomatic properties or computational bounds to be transferred across domains. In particular, lower bounds on the complexity for winner determination or manipulative interference are inherited upwards, while upper bounds are inherited downwards. Similarly, the violation of an axiom transfers upwards, while its satisfaction transfers downwards.

Having this abstract model in mind as a common umbrella for the studied, specialized frameworks (modeled by restricting the domains $\mathcal{X}$ and $\mathcal{Y}$ accordingly), we are set to formulate five goals this thesis intends to acquire. In particular, this thesis pursues to (partially) answer five types of research questions. Each is discussed and motivated in the subsequent sections to understand the relevance of the respective research goals and their implications for practical use. However, the intention is not to find comprehensive answers to these universal questions. Instead, our aim is to contribute valuable insights with practical relevance, to shed light on a complex and manifold research field.

[41] Exhaustiveness can also be enforced artificially by considering rules which fill up unexhausted bundles.

3.1 Axiomatic Analysis

The first set of research questions aims to uncover a deeper understanding of the behavior and underlying constraints associated with voting rules. This is mostly done by designing and studying desirable axiomatic properties a rule may or may not satisfy. Undeniably, this is one of the most crucial questions to address and thus deserves an extensive discussion. In order to derive a collective decision from (possibly contradicting) individual preferences or opinions, we are less interested in what rule to actually use, but rather in whether an outcome is in the best interest of the voters.

Let us give just a few examples of desirable axiomatic properties a voting rule or its outcome(s) should satisfy. We might axiomatically quantify *fairness* by requiring that a rule should always act *anonymous* or *neutral* [43] (i.e., treating voters or alternatives equally), or *proportional* [5, 6, 45] (i.e., ensuring a proportional representation of any preference group in an outcome relative to its size). For *safety*, a rule should be robust against *manipulation* [85, 154] (i.e., the strategic change of a voter's preference) or *control* [11] (i.e., the strategic change of an election's specifications). Similarly, we may quantify *performance* by requiring that the outcome of a rule should be *exhaustive* [6] (i.e., an additional alternative cannot be added without violating feasibility) or *competitive* [82] (i.e., if voters' satisfaction with an outcome can be measured by some metric, a rule should guarantee some kind of minimum satisfaction). Although not explicitly studied in this thesis, note that there are many *characterization* results across related literature, stating that a (set of) rule(s) can be well-defined by a set of satisfied axioms.[42]

We can tackle an axiomatic analysis from two directions: *(i)* From the perspective of modeling, we may act result-oriented by designing (or choosing) a rule based on a set of axiomatic properties, which are desired for a given real-world scenario. In case such a rule does not exist due to incompatible axioms, we may approximate our modeling goals by trading off properties. For illustration, assume there is no well performing rule which is fair and safe (according to predefined properties). Then a relaxation of the requirements might allow for a fair rule with performance guarantee, which is *almost* safe. *(ii)* Second, having a rule in place, it is reasonable to either verify its justification by investigating whether desired criteria are met, or extend the study on yet neglected properties. This is particularly important for rules relying on a metric, stating how *good* a solution is. For example, a rule that maximizes a scoring function (which is assumed to accurately reflect the voters' satisfaction with an outcome) has a justified efficiency per se, while an axiomatic analysis may reveal at what cost.

Note that studying axiomatic properties in the context of our global model may have implications across all three considered research fields. For example, if a rule violates an axiom for multiwinner elections, the violation is implied for its generalized variants in participatory budgeting and judgment aggregation, too. Vice versa, satisfied criteria for

[42] E.g., quota rules following Dietrich and List [66] or the median rule by Nehring and Pivato [133].

an (asymmetric) judgment aggregation rule inherit downwards by choosing a multiwinner or budgeting constraint. Overall, we summarize these types of research goals, related to the axiomatic analysis of voting rules, compactly in Question Q1 as follows:

Question Q1 (Axiomatic Analysis). *What are desirable axiomatic properties an outcome or any rule F should satisfy in a domain-specific research field? How does either a fixed rule F or a class of such rules behave axiomatically? Which axiomatic properties are compatible with one another, i.e., is there at least one rule that satisfies multiple predefined axioms simultaneously?*

Understanding the axiomatic behavior of voting rules is not only of pure theoretical interest. In fact, its implications can significantly contribute to assessing potential risks and chances of improvement for practical applications. First, focusing on the desired result (i.e., predetermining desirable criteria) helps finding a suitable rule, which acts as requested. Second, designing and evaluating a rule axiomatically may improve transparency, resulting in a more justifiable outcome. Hence, understanding *why* a result was chosen can lead to a more credible voting process with a higher acceptance by participants.[43] Third, knowing that a rule violates desired criteria can indicate potential drawbacks. In some cases this may pave the way for creative opportunities to bypass said violation. For example, a voting rule prone to strategic changes of given parameters may be safe if all parameters are fixed in advance.

3.2 Complexity of Winner Determination

Having a rule in place, we are interested in the computational complexity required to derive a *winning* outcome, based on the voters' preferences. While it might be practically sufficient to distinguish between *efficiently computable* and *computationally hard* rules, our aim is to demonstrate which aspects of winner determination contribute to how much complexity. In particular, knowing which parameters of an underlying election render a rule hard (e.g., the number of participants) might allow to escape intractability by parameterization.[44] Primarily, we focus on yet another aspect: As we mainly study irresolute rules, we consider a more diverse study on the complexity of winner determination problems (in contrast to simply classifying considered rules as hard or efficiently computable). This results in more evolved insights into the complexity for different aspects of winner determination. In particular, as the outcome of an irresolute rules is a set, we study how hard it is to decide whether an alternative is appearing *possibly* in an outcome (i.e., in at least one) or *necessarily* (i.e., in all).[45]

[43] To enhance acceptance by understanding, both Cailloux and Endriss [49] and Peters, Procaccia, Psomas, and Zhou [143] use a series of simple axiomatic arguments to *explain* the result of a voting rule.

[44] De Haan [93] discusses parameterized complexity and its applications to judgment aggregation.

[45] The existential (universal) variant is also known as *credulous* (*skeptical*) outcome determination [75].

By our global model, lower bounds on the complexity of winner determination are inherited upwards and upper bounds downwards. We summarize these research goals regarding the complexity of winner determination in Question Q2 as follows:

Question Q2 (Winner Determination)**.** *What is the computational complexity associated with questions related to the outcome of a rule F?*

With respect to practical usage, we hope for rules that adequately meet our modeling goals while being efficiently computable. Sadly, this is often not the case. Knowing *if* a rule is computationally hard allows for a mathematically grounded discussion on why a rule might be unsuitable for a given use-case. In turn, we may relax desired modeling requirements to receive tractability. By deriving insights in *why* a rule is hard to compute, we may find it more effective to adjust the overall election instead of the underlying rule in order to achieve our goals. For example, determining an optimal bundle in participatory budgeting is hard for cost-based satisfaction [166]. Yet, if we adjust each project's cost upwards to the next thousandth of the budget limit, using the same rule, we can efficiently compute an outcome as a reasonable approximation (see Kellerer, Pferschy, and Pisinger [104] for a pseudo-polynomial implementation).

3.3 Complexity of Manipulative Interference

Sadly, most voting rules are prone to strategic changes by various forms of *manipulative interference*. Following a famous impossibility result by Gibbard [85] and Satterthwaite [154] (based on ordinal preferences), all voting rules satisfying a short list of very basic criteria can be manipulated by voting untruthfully. As a way to limit manipulative interference in practice, we study whether computing a manipulator's strategy is computationally (too) demanding. Note that many forms of manipulative interference can also be interpreted from a more optimistic perspective: As questions about the *robustness* of an alternative against any changes within a certain range. More precisely, even if not all parameters of an election can be fixed in advance (e.g., by considering random noise in the votes [32]), we might already determine that some alternatives must clearly appear in any outcome, while others appear at least in some cases.

Manipulative interference has been studied extensively in various forms depending on the underlying framework and especially the use-case.[46] Informally, we often classify manipulative interference into three distinct categories: For questions regarding *manipulation* [12, 73] we are interested in whether a voter (or sometimes a group of voters [37]) can change her vote strategically to align the resulting outcome further with her preference (e.g., in a constructive variant to include a preferred set of alternatives or in a destructive variant to exclude). Relating to our general model, manipulation refers to the alteration

[46]For an overview, including examples in the fields of preference aggregation and judgment aggregation, we refer to the textbook by Rothe [152].

of one fixed entry of the input list X. For *bribery* [79, 18], an agent may bribe a limited number of voters to alter their votes in an attempt to change the outcome in favor of the briber. Formally, a fixed number of entries of X can be altered by a valid bribery attempt. Lastly, for questions revolving around *control* [11, 17, 16], we assume someone in charge of the electoral process has the power to alter the election's structure to some degree. This can take various forms, for example, by in- or excluding voters from submitting their ballots. As a use-case oriented example in participatory budgeting, a valid control attempt might involve changing the overall budget limit or some alternative's cost function.

In this thesis we focus mainly on *structural* changes resulting from combinatorial aspects (i.e., changing $\mathcal{Y}$ or F) and consider problems regarding manipulation or bribery (i.e., changing X) only from a more general perspective. We summarize our research goals related to the complexity of manipulative interference in Question Q3 as follows:

> **Question Q3** (Manipulative Interference)**.** *Is it possible to change the outcome in favor of a manipulator by slightly adjusting either $\mathcal{Y}$ or F (or X) to a given extent? If this is the case, how computationally demanding is it to find a successful strategy for changing the outcome favorably?*

Finding answers to these questions allows us to better evaluate the risk for a decision-making process of being strategically influenced. If we are able to efficiently compute an outcome, while relevant manipulative actions are hard to compute, the threat of being manipulated is rather small (for large enough elections). If this is not the case, we may at least hope that questions regarding manipulative interference are computationally harder to decide than winner determination itself. Lastly, especially for problems regarding the combinatorial structure of a decision-making process, we may often bypass those types of control by fixing all relevant parameters in advance. Identifying respective complexities first, we know for which parameters this is the most relevant.

3.4 Relationships Between Rules

Although the rules we consider can be interpreted as explicit functions that map any valid input to a corresponding output, two independently formulated descriptions of the same function may differ strongly. To give an example, consider the sequential majority rule, introduced in Subsection 2.4.2. For the resolute variant we can reformulate its sequential description into a maximizing rule, where each formula supported by the majority is weighted by decreasing powers of two (based on the processing order). Hence, results obtained from individually studied rules that are shown to be identical can be unified. When rules were formulated for different domains (e.g., multiwinner elections and judgment aggregation), this allows connecting the respective fields more closely together. In other cases, rules may relate as generalization and refinement to one another (if there is always an inclusion relationship between outcomes for identical inputs).[47] Linking those rules allows for a transfer of results related to computation or axiomatic properties. Lastly, in rare cases it is not obvious that (similar) rules are indeed different, which may impact the result in terms of the desired criteria negatively. For example, although for the median rule in judgment aggregation we may either minimize the hamming distance or maximize the number of approved issues, a similar approach does only hold for the egalitarian median rule if each voter approves the same number of issues. While in standard judgment aggregation this is always the case (i.e., each individual judgment must approve exactly half of the agenda items), in an asymmetric setting we may consider two rules that are similar but fundamentally different.[48]

These examples illustrate that uncovering possible relationships between rules can be a complex but rewarding task. We summarize our aim for a deeper understanding of the interaction among individually studied voting rules in Question Q4 as follows:

> **Question Q4** (Relationships Between Rules)**.** *How do aggregation rules relate to one other? How can we meaningfully generalize rules to different domains? Can a rule be seen as a refinement of another rule? Are there non-obvious (dissimilarly defined) coinciding rules, either in a closed research field or across related disciplines?*

Revealing relationships between rules can yield relevant implications for real-world applications. Indeed, unifying research results by identifying related rules may allow for better educated and more fine-grained selections of suitable voting rules. From a computational point of view, it is of particular interest to identify complexity gaps for winner determination problems among coinciding rules of different domains. This could hold a deeper insight in *what* renders a related problem hard.[49] In case there is no complexity gap, the upside from switching to a more expressive framework comes without a (significant) computational trade-off, which is discussed extensively in Chapter 8.

3.5 Ballot Design

Going one step beyond the limits of simple approval ballots, our aim is to explore how generalizing the input format slightly may positively impact the overall decision-making process, without putting too much cognitive burden on the voters. Let us motivate the choice of a ballot format from a broader perspective: To derive a collective decision from individual preferences, we are in between two extremes. On the one hand, if voters are able to express their preferences in detail, a potentially optimized decision comes with a high complexity. That is, in communication of the voters' opinions, as well as the

[47] For example, the maxcard subagenda rule is a refinement of the maximum subagenda rule [115, 118].

[48] To see that egalitarian rules (based either on the hamming distance or the number of approved alternatives) do not coincide, consider a profile with two voters, respectively approving none or all alternatives.

[49] In a related approach, de Haan [91] identified judgment aggregation constraint types, such that winner determination for a variety of rules become tractable.

evaluation by a suitable mechanism. On the other hand, if we rely on a limited, but easy to elicit, ballot format, simplicity comes with a loss of information. This in turn may result in an uninformed decision, disregarding the voters' *actual* preferences. Having this trade-off in mind, there are many established ballot formats for different applications, such as approving, ranking, or rating single alternatives. Yet, those formats do not allow for expressing logical dependencies between different alternatives in an outcome. While this can be easily implemented by approving, ranking, or rating committees instead of isolated candidates, the ballot's size might become exponential in the number of alternatives.

Hence, in an application-oriented approach, an important task is to find a sweet spot between both extremes. That is, finding a ballot format which can realistically (enough) capture the voters' preferences, while being simple (enough) to cast. We investigate this research direction, by studying slight generalizations of basic approval ballots and summarize our last research goal in Question Q5 as follows:

> **Question Q5** (Ballot Design)**.** *To what extent can a slightly more general ballot format, allowing participants to express their opinions in a more evolved fashion (than by simple approvals), yield better outcomes?*

By formally introducing new ballot formats, we extend the realm of choices for real-world applications, while a subsequent theoretical evaluation of those formats can allow for an educated decision based on the use-case. In particular, for practical use we can derive multiple implications based on our theoretical results. If a decision-making process can be improved significantly by allowing for a more expressive ballot format without adding too much cognitive burden on the voters, the upside from switching to a less studied ballot format might outweigh potential downsides. In situations where the benefit of more evolved preferences is only marginal, choosing a more common ballot format along with a well-studied voting rule might be more effective. This is of particular interest when there is a metric present to quantify how *good* an outcome is based on the voters' preferences. By default, this generally holds for scoring rules, which are studied extensively throughout this thesis.

CHAPTER 4

IRRESOLUTE APPROVAL-BASED BUDGETING

In this chapter, we revisit participatory budgeting with approval-based preferences, to generalize established results to an irresolute context. Overall, we reinterpret previously considered (resolute) rules and axiomatic properties, and introduce a new class of hybrid rules as approximation with performance guarantee.

4.1 Summary

In this work, we extend the axiomatic analysis on voting rules for participatory budgeting to an irresolute setting. We build on the study by Talmon and Faliszewski [166], who introduced a formal framework for participatory budgeting with approval-based preferences. For a more valuable insight into our contribution, let us first recap the model by Talmon and Faliszewski. The authors use three reasonable approaches to model the satisfaction of a single voter with a given outcome (generalized from multiwinner elections), and then use three different aggregation rules, aiming to optimize the voters' overall satisfaction. The modular setup yields a total of nine voting rules, which are interpreted as resolute rules, using a tie-breaking scheme if necessary. In addition, they present desirable axioms any budgeting method should satisfy, that emerge from considering a budgeting constraint. All considered axioms relate to the robustness of voting rules. For example, an item in the outcome should not be excluded by a reduced cost for implementation.

We revisit the given framework, reinterpreting the rules to be irresolute (assuming parallel-universe tie-breaking). We begin by exploring computational aspects, showcasing that two rules studied independently actually coincide. Furthermore, we show that for greedy rules, which can be interpreted as approximations, there is no performance guarantee (i.e., the voters' overall satisfaction can be inversely proportional to the satisfaction of an optimal solution). To circumvent that issue, we introduce a simple class of *hybrid greedy* *rules* by combining two slightly different approaches, yielding a constant approximation factor. In particular, for performance guarantee, we show that our introduced hybrid rules coincide with an approximation for the *budgeted maximum coverage problem*, studied by Khuller, Moss, and Naor [106]. As an intermediate result, we connect our rules based on maximization to restricted domains of the budgeted maximum coverage problem.

Finally, as there can be multiple outcomes due to irresoluteness, we redefine related axioms in an existential fashion[50] and study the properties of all introduced rules. Notably, we see that performance guarantee for hybrid rules comes with a trade-off, as those rules perform poorly axiomatically. Formal proofs, which were omitted from our publication [25] due to space constraints, are supplemented in Appendix A.1.

4.2 Reflection on Initial Research Goals

In this section, let us reflect on how this article's key results contribute to answering four of our initial research questions, motivated in Chapter 3. Most extensively, we addressed Question Q1, as our axiomatic analysis provides valuable insights into how voting rules behave in the domain of participatory budgeting in an irresolute context. We partially dealt with Question Q2 by introducing a new class of hybrid rules, capable of identifying a single outcome in polynomial time. Although we did not study Question Q3 through the lens of computational complexity, the satisfaction of a particular axiom implies robustness of a voting rule against appropriate (strategical) structural changes to a budgeting campaign. In contrast, the violation of an axiom may allow for manipulative interference, which we study more comprehensively in Chapter 5. Lastly, by *(i)* identifying coinciding rules within a shared domain and *(ii)* linking all three max rules to (restricted domains of) the budgeted maximum coverage problem, we addressed Question Q4.

4.3 Publication

This work has been published and presented as an extended abstract at the *19th International Conference on Autonomous Agents and Multiagent Systems*.

[25] D. Baumeister, L. Boes, and T. Seeger. "Irresolute Approval-based Budgeting". In: *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*. IFAAMAS, 2020, pp. 1774–1776

[50] Similarly, Rey, Endriss, and de Haan [149] generalize by considering a universal quantifier instead.

4.4 Personal Contribution

The work on this article was initiated by Tessa Seeger's impressive bachelor's thesis, where Theorem 2.1 and the axiomatic analysis for all rules except for the hybrid greedy rules were obtained and published in a preliminary version. The conception and writing of this article was conducted jointly with Dorothea Baumeister. The remaining technical results — designing and analyzing the hybrid greedy rules axiomatically, as well as Propositions 3.1 and 3.2 — were contributed by me.

Irresolute Approval-based Budgeting

Extended Abstract

Dorothea Baumeister, Linus Boes, Tessa Seeger
Institut für Informatik
Heinrich-Heine-Universität Düsseldorf
{d.baumeister,linus.boes,tessa.seeger}@uni-duesseldorf.de

ABSTRACT

In participatory budgeting, citizens can take part in the decision on which projects a city should spend money. Formally, the input is a set of items, each having a certain cost, while agents can express their preferences. The task is to choose a set of items respecting a given bound. Recently Talmon and Faliszewski [10] introduced a framework for budgeting based on approval votes. This paper revisits the introduced methods axiomatically from an irresolute point of view, especially showing that two of the proposed methods coincide. The study is complemented by approximation results.

KEYWORDS

participatory budgeting; voting; approximation

ACM Reference Format:
Dorothea Baumeister, Linus Boes, Tessa Seeger. 2020. Irresolute Approval-based Budgeting. In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, May 9–13, 2020*, IFAAMAS, 3 pages.

1 INTRODUCTION

In participatory budgeting (PB), citizens are directly involved in the process of collective decision making at municipal or even global level. More precisely, the participants may express their preferences, by voting on which multitude of proposals (at non-uniform cost) public funds should be spent. We consider proposals either fully funded or rejected, in contrast to e.g. Freeman et al. [6], where funds may be divided non-discretely. Due to the rise of digital democracy, such processes are relevant to a large group of people, and the formal framework may be used to make decisions in different contexts. Participatory budgeting may be interpreted as a generalization of multiwinner elections, where each alternative occupies a fixed amount of seats. Following this generalization, there are various approaches to model voters' preferences. Goel et al. [7], Fluschnik et al. [5], and Benade et al. [4] consider assigning each alternative a utility, while Lu and Boutilier [9] consider participation by ranking alternatives. Benade et al. [3] evaluate multiple approaches in an empirical study. An overview of current research on participatory budgeting from a computational social choice perspective is given in the book chapter by Aziz and Shah [2].

Many cities, like Paris for example, that actually conduct PB rely on approval votes, where the voters may simply vote for some (possibly restricted) subset of the alternatives. We will also use approval-based preferences over the set of alternatives as it was proposed by Aziz et al. [1]. In particular, we will expand the framework by Talmon and Faliszewski [10], where satisfaction functions are used to evaluate how good a budget represents voters' preferences.

We contribute by interpreting the given model for irresolute budgeting scenarios, where the result may be a set of feasible winning budgets instead of one distinct bundle. Hence, we generalize studied axioms to irresolute budgeting methods by slightly altering their definition. Notably, we will show that two methods introduced by Talmon and Faliszewski [10] actually coincide irrespective of the used tie-breaking methods. Furthermore, we interpret rules that rely on greedy approaches as approximations and study their performance in contrast to optimal solutions.

The choice to focus on irresolute rules is motivated by the practical application of this framework. Although unique solutions are desirable, ties occur naturally, and breaking ties without participants' consent is at risk of losing either transparency or credibility. For preserving democratic deliberation, it is reasonable to assume that the tie-breaking will be made by the municipality. In extreme cases, breaking ties differently may result in disjunct budgets, which indicates the power, that the tie-breaking authority has. This might naturally lead to a conflict of interest when tie-breaking is not further specified and a possible waste of resources when tie-breaking is fixed priorly. Overall, most real-world campaigns are conducted in multiple stages to guarantee a favorable and realizable outcome. Hence, reaching a consensus that reflects the communities' preferences more precisely by adding a top-layer (i.e. deliberately breaking ties) is exactly in the spirit of PB.

2 PRELIMINARIES

We adopt the framework by Talmon and Faliszewski [10]. Hence, we consider a budgeting scenario as quadruple $E = (A, V, c, \ell)$, consisting of m items $A = \{a_1, \ldots, a_m\}$, a function $c\colon A \to \mathbb{N}$ assigning a cost to each item, n voters $V = \{v_1, \ldots, v_n\}$ each balloting with a set of approved items $A_v \subseteq A$ for $v \in V$, and a budget limit $\ell \in \mathbb{N}$. We denote the set of items from a budget $B \subseteq A$, also approved by voter v as $B_v = A_v \cap B$. In this paper we use composite budgeting methods $\mathcal{R}_f^r$ as defined by Talmon and Faliszewski [10], but interpret them as irresolute procedures. Hence, each method $\mathcal{R}_f^r$ takes any budgeting scenario E as input and outputs a nonempty set of winning budgets $\mathcal{R}_f^r(E) \subseteq 2^A \setminus \{\emptyset\}$. This is done by applying a budgeting rule r, respecting a satisfaction function $f : 2^A \times 2^A \to \mathbb{N}$. We adopt proposed satisfaction functions f, also introduced by Talmon and Faliszewski [10], to derive the satisfaction of a voter from her approval ballot, focussing on either the quantity $f(A_v, B) = |B_v|$, the cost of approved items that are budgeted $f(A_v, B) = c(B_v)$

Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 9–13, 2020, Auckland, New Zealand.

(where slightly abusing notation it holds $c(B) = \sum_{b \in B} c(b)$ for every bundle $B \subseteq A$)), or the presence of at least one approved item in the budget $f(A_v, B) = \mathbb{1}_{|B_v|>0}$.

Similarly to the satisfaction functions, we adopt the definition of max rules, greedy rules, and proportional greedy rules, additionally we introduce hybrid greedy rules. The formal definition with respect to a given satisfaction function f is as follows.

Max rules ($\mathcal{R}_f^m$): $\mathcal{R}_f^m(E) = \arg\max_{B \subseteq A} \sum_{v \in V} f(A_v, B)$, while respecting the budget limit $\sum_{b \in B} c(b) \leq \ell$.

Greedy rules ($\mathcal{R}_f^g$): Starting with $B = \emptyset$ iteratively extend B by $a \in A \setminus B$, maximizing $\sum_{v \in V} f(A_v, B \cup \{a\})$, such that $\sum_{b \in B} c(b) \leq \ell$.

Proportional Greedy rules ($\mathcal{R}_f^p$): Similar to the greedy rule, maximize $(\sum_{v \in V} f(A_v, B \cup \{a\}) - \sum_{v \in V} f(A_v, B))/\ c(a)$ iteratively, starting with $B = \emptyset$.

Hybrid Greedy rules ($\mathcal{R}_f^h$): Arbitrarily select $B_g \in \mathcal{R}_f^g(E)$ and $B_p \in \mathcal{R}_f^p(E)$ and output the budget with maximum satisfaction $\arg\max(\sum_{v \in V} f(A_v, B_g), \sum_{v \in V} f(A_v, B_p))$ as $\mathcal{R}_f^h(E)$.

We interpret all three greedy variants as irresolute rules, by considering every budget as winning, that may result from breaking the ties in each iteration. We now show that two of the considered budgeting methods coincide.

Theorem 2.1. *$\mathcal{R}^g_{|B_v|}$ and $\mathcal{R}^p_{c(B_v)}$ are equivalent, i.e. they always output the same set of winning budgets.*

Proof. First note that both above binary satisfaction functions f respectively map to an unary function f', where $f(A_v, B) = f'(B_v)$. While maximizing iteratively we may ignore constant factors $f'(B_v)$ carried over from previous iterations, assuming f' is additive. Hence in each iteration, the greedy rule $\mathcal{R}_f^g$ is selecting an item a maximizing $\sum_{v \in V} f'(A_v \cap \{a\})$ while the proportional greedy rule $\mathcal{R}_f^p$ selects item a maximizing $\sum_{v \in V} f'(A_v \cap \{a\})/c(a)$. Further for any f' with $f'(\emptyset) = 0$ it follows that

$$\sum_{v \in V} f'(A_v \cap \{a\}) = |\{v \in V \mid a \in A_v\}| \cdot f'(\{a\}).$$

Note that $|B_v|$ and $c(B_v) = \sum_{b \in B_v} c(b)$ are indeed additive and map to zero for $B_v = \emptyset$. By applying above implications, we conclude that both $\mathcal{R}^g_{|B_v|}$ and $\mathcal{R}^p_{c(B_v)}$ iterate by selecting item a maximizing the value $|\{v \in V \mid a \in A_v\}|$, since $|\{a\}| = c(\{a\})/c(a) = 1$. □

This theorem holds irrespective of the used tie-breaking, since in each iteration the same items may be chosen. Hence, also in the setting of Talmon and Faliszewski [10], both rules are equivalent.

3 APPROXIMATION AND PROPERTIES

We interpret given greedy approaches as approximations and study their performance in contrast to optimal solutions (i.e. max rules).

Proposition 3.1. *$\mathcal{R}^g_{|B_v|}$, $\mathcal{R}^g_{\mathbb{1}_{|B_v|>0}}$, $\mathcal{R}^p_{c(B_v)}$, $\mathcal{R}^p_{|B_v|}$ and $\mathcal{R}^p_{\mathbb{1}_{|B_v|>0}}$ do not have a constant approximation factor.*

This can be shown by counter examples, where the approximation factor is inversely proportional to the budget limit. In contrast, it can be shown that the newly introduced hybrid greedy rules have a $(1 - 1/\sqrt{e})$-approximation.

Proposition 3.2. *For all three satisfaction functions f considered here and every $B_m \in \mathcal{R}_f^m(E)$ and $B_h \in \mathcal{R}_f^h(E)$, it holds that $\sum_{v \in V} f(A_v, B_h)/ \sum_{v \in V} f(A_v, B_m) \geq 1 - 1/\sqrt{e}$.*

Above proposition follows by a similar $(1 - 1/\sqrt{e})$-approximation due to Khuller et al. [8] and the insight, that each max rule can be modeled as a special case of the budgeted maximum coverage problem.

Now, we recap some of the proposed axiomatic properties by Talmon and Faliszewski [10] to study them irrespective of the used tie-breaking rule. Hence, we slightly adapt the properties in order to handle irresolute rules.

Definition 3.3. Let $E = (A, V, c, \ell)$ be a budgeting scenario with $B \in \mathcal{R}(E)$. The following axiomatic properties are satisfied by a budgeting rule $\mathcal{R}$, if for every modified budgeting scenario E' (as defined below) there exists a budget $B' \in \mathcal{R}(E')$, meeting a requirement as defined:

Limit Monotonicity: For $E' = (A, V, c, \ell + 1)$, where for all $a \in A$ it holds $c(a) \neq \ell + 1$, we require $B \subseteq B'$.

Discount Monotonicity: For $b \in B$ and $E' = (A, V, c', \ell)$ with $c'(a) = c(a)$ for every $a \in A \setminus \{b\}$, and $c'(b) = c(b) - 1$, we require $b \in B'$.

Splitting Monotonicity: For $a \in B$ and every E', where a is split into a set of items A', an extended cost function satisfying $c(a) = \sum_{a' \in A'} c(a')$, and exactly those voters approving a, approve all items in A', we require $A' \cap B' \neq \emptyset$.

Merging Monotonicity: Let $A' \subseteq B$, such that for each $v \in V$ it holds either $A_v \cap A' = \emptyset$ or $A' \subseteq A_v$. For E', where A' is merged into a new item a, an extended cost function satisfying $c(a) = \sum_{a' \in A'} c(a')$, and the voters approving a are exactly those who approved A', we require $a \in B'$.

Our results are consistent with those for resolute methods and summarized in Table 1.

Table 1: Axiomatic properties of budgeting methods. Results are generalized from Talmon and Faliszewski [10]. Deviations are marked by ▲ (see Theorem 2.1), new results by ♣.

$\mathcal{R}_f^r$	m	g	p	h	m	g	p	h	m	g	h
		$\lvert B_v \rvert$				$\mathbb{1}_{\lvert B_v \rvert>0}$				$c(B_v)$	
Limit M.	x	x	x	x♣	x	x	x	x♣	x	x	x♣
Discount M.	✓	✓▲	✓	x♣	✓	✓	✓	x♣	x	x	x♣
Splitting M.	✓	✓	✓	✓♣	✓	✓	✓	x♣	✓	x	x♣
Merging M.	x	✓▲	x	x♣	✓	✓	x	x♣	✓	✓	x♣

When considering resolute methods, there are underlying assumptions, which might not be resolved easily. Some of the considered axioms might be violated if the tie-breaking scheme depends on the cost or the total quantity of items budgeted. Even the application of linear mechanisms might not be trivial, as splitting or merging items might interfere with the order.

ACKNOWLEDGMENTS

This work was supported in part by a grant from the NRW Ministry of Culture and Science.

REFERENCES

[1] Haris Aziz, Barton Lee, and Nimrod Talmon. 2018. Proportionally Representative Participatory Budgeting: Axioms and Algorithms. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 23–31.

[2] Haris Aziz and Nisarg Shah. 2020. Participatory Budgeting: Models and Approaches. In *Pathways between Social Science and Computational Social Science: Theories, Methods and Interpretations*, T. Rudas and P. Gabor (Eds.). Springer. To appear.

[3] Gerdus Benade, Nevo Itzhak, Nisarg Shah, Ariel Procaccia, and Ya'akov Gal. 2018. Efficiency and Usability of Participatory Budgeting Methods. (2018). Manuscript.

[4] Gerdus Benade, Swaprava Nath, Ariel Procaccia, and Nisarg Shah. 2017. Preference Elicitation for Participatory Budgeting. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*.

[5] Till Fluschnik, Piotr Skowron, Mervin Triphaus, and Kai Wilker. 2019. Fair Knapsack. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, Vol. 33. 1941–1948.

[6] Rupert Freeman, David Pennock, Dominik Peters, and Jennifer Wortman Vaughan. 2019. Truthful Aggregation of Budget Proposals. In *Proceedings of the 2019 ACM Conference on Economics and Computation (EC '19)*. ACM, New York, NY, USA, 751–752.

[7] Ashish Goel, Anilesh Krishnaswamy, Sukolsak Sakshuwong, and Tanja Aitamurto. 2015. Knapsack Voting. *Collective Intelligence* 1 (2015).

[8] Samir Khuller, Anna Moss, and Joseph Seffi Naor. 1999. The Budgeted Maximum Coverage Problem. *Information processing letters* 70, 1 (1999), 39–45.

[9] Tyler Lu and Craig Boutilier. 2011. Budgeted Social Choice: From Consensus to Personalized Decision Making. In *Twenty-Second International Joint Conference on Artificial Intelligence*.

[10] Nimrod Talmon and Piotr Faliszewski. 2019. A Framework for Approval-based Budgeting Methods. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, Vol. 33. 2181–2188.

CHAPTER 5

COMPLEXITY OF MANIPULATIVE INTERFERENCE IN PARTICIPATORY BUDGETING

In this chapter, nested in the scope of participatory budgeting, we study the computational complexity of *(i)* winner determination by complementing previous results with a more diverse study on possible and necessary winner variants, and *(ii)* manipulative interference from a generic point of view and in detail for two budgeting-specific control problems.

5.1 Summary

In this work, we use computational complexity theory to examine questions related to winner determination and manipulative interference for approval-based participatory budgeting. This research has been initiated by Talmon and Faliszewski [166], who study these question in its simplest form. For resolute budgeting methods, they identified *(i)* which rules are hard to compute and *(ii)* which rules satisfy axiomatic properties that arise from the combinatorial structure of a participatory budgeting campaign. A violation of the latter implies a vulnerability to electoral control. As an intermediate step we [25] generalized respective results to an irresolute setting, as discussed in Chapter 4. Overall, we extend those results by studying both winner determination and strategic aspects in a structured and more fine-grained fashion.

For winner determination, we study the computational complexity of a total of four decision problems. In particular, we extend the study on irresolute winner determination by possible (and necessary) winner variants, where the question is, whether at least one (respective all) winning bundle(s) contain(s) a predefined subset of items. For rules maximizing an efficiently computable satisfaction function, we present a general upper bound scheme to derive (not necessarily tight) upper bounds for all decision problems simultaneously. Those results are complemented by providing a matching lower bounds for the only hard to compute rule we consider.

To study manipulative interference from a general perspective, we introduce alteration functions, that map from a given budgeting scenario to a set of possible scenarios after the alteration (e.g., the strategic change of a voter's ballot). For a given budgeting method $\mathcal{R}_s$, we study the following decision problem, formalizing the question, whether a manipulator can include at least k items from a subset of items $B_{\heartsuit} \subseteq A$ into a winning bundle, by changing the given campaign by a valid alteration (due to a given alteration function f). In a destructive variant, the goal is to prevent items from being in a winning bundle.

CONSTRUCTIVE-$\mathcal{R}_s$-MANIPULATIVE-INTERFERENCE (C-$\mathcal{R}_s$-MI)	
Given:	A budgeting scenario E, a set of items $B_{\heartsuit}$, an integer k, and an alteration function f.
Question:	Is there a budgeting scenario $E' \in f(E)$, such that there is a winning bundle $B \in \mathcal{R}_s(E')$ with $\lvert B_{\heartsuit} \cap B\rvert \geq k$?

After identifying trivial upper bounds, derived from the complexity for winner determination, we explicitly study control problems that rely on the combinatorial structure given in participatory budgeting. In particular, we study two types of alteration functions: By setting either *(i)* the budget limit; or *(ii)* a specific item's cost for a given budgeting campaign to some value in a given interval. For the latter case, we only study whether the specific item itself can be in- or excluded by an alteration.

For most of the considered control problems, we present polynomial-time computable algorithms, hinting that the considered parameters should be fixed in advance to avoid manipulative interference occuring in practice.

5.2 Reflection on Initial Research Goals

This article contributes significantly to answering two of our initial research questions, introduced in Chapter 3. We addressed Question Q2, by *(i)* initiating a more diverse study on the computational complexity for winner determination in an irresolute participatory budgeting setting, *(ii)* providing a general proof scheme for upper bounds, and *(iii)* establishing tight bounds for the only computationally hard rule we consider. Furthermore, we addressed Question Q3, by considering alteration functions to present a generic approach to study manipulative interference in participatory budgeting through the lens of computational complexity. Along with an upper bound proof scheme, we studied the lower-bound complexity for two suitable control problems in both constructive and destructive variants.

5.3 Publication

This work has been published and presented as a full paper at the *7th International Conference on Algorithmic Decision Theory*.

[22] D. Baumeister, L. Boes, and J. Hillebrand. “Complexity of Manipulative Interference in Participatory Budgeting”. In: *Proceedings of the 7th International Conference on Algorithmic Decision Theory*. Springer. 2021, pp. 424–439

5.4 Personal Contribution

The work on this article was initiated by Johanna Hillebrand’s remarkable bachelor’s thesis, where questions related to winner determination and manipulative interference were initially studied and published in a less general way. In particular, Theorem 5 was already shown in Johanna Hillebrand’s thesis in a similar way. All the remaining technical results required more evolved reasoning and were contributed by me. The overall conception and writing of this article was conducted jointly by Dorothea Baumeister and me.

Complexity of Manipulative Interference in Participatory Budgeting

Dorothea Baumeister, Linus Boes✉, and Johanna Hillebrand

Heinrich-Heine-Universität Düsseldorf, Düsseldorf, Germany
{d.baumeister,linus.boes,johanna.hillebrand}@uni-duesseldorf.de

Abstract. A general framework for approval-based participatory budgeting has recently been introduced by Talmon and Faliszewski [17]. They use satisfaction functions to model the voters' agreement with a given outcome based on their approval ballots. We adopt two of their satisfaction functions and focus on two types of rules. That is, rules that maximize the overall voters' satisfaction and greedy rules that iteratively extend a partial budget by an item that maximizes the satisfaction in each incremental step. An important task in participatory budgeting is to study different forms of manipulative interference that may occur in practice. We investigate the computational complexity of different problems related to determining the outcome of a given rule and give a very general formulation of manipulative interference problems. A special focus is on problems dealing with a varying cost of the items and a varying budget limit. The results range from polynomial-time algorithms to completeness in different levels of the polynomial hierarchy.

1 Introduction

Participatory budgeting is often implemented as a mean of making democratic decisions. Thus, citizens can usually express their opinions on how a portion of a city's budget should be distributed in such a process, sometimes making new suggestions on which projects could be realized as well. The first implementation of participatory budgeting can be found in Porto Alegre (Brazil) in 1989 as an attempt by the Workers Party to break with traditionally authoritarian public policies (see Sintomer et al., [15]). Starting here the idea spread around the world, taking different forms and magnitudes regarding size, budget, and other factors. As described by Cabannes [5], integrating participatory budgeting into a city's form of government has yielded several positive effects. These range from an increased accountability of politicians, due to the collective will being rather visible, to directing larger parts of the city's budget towards education, health care, infrastructure, and childcare. The different stages of a participatory budgeting cycle are described by Aziz and Shah [3]. These stages include the division into different districts, the determination of the total available budget, the emergence of project proposals, deliberation steps, and finally the voting stage. While Rey et al. [14] study multiple stages in one model, we will solely focus on the last step in this paper. Here, the citizens express their preferences regarding the projects they want the budget to be spent on. A participatory budgeting method then aggregates these votes in order to reach a decision about which projects will be realized. Here we assume that each project is either fully funded or not at all. So, in this last step we have a fixed set of projects, each associated with a cost, and we have an overall budget limit. The total cost of the funded projects must be within this limit. A crucial point is how the preferences of the voters are expressed, as a decision between expressiveness and compactness of the presentation has to be made. We follow a very simple approach by assuming approval ballots, where every voter chooses for every project whether she thinks that it should be funded or not. These individual votes are independent of the budget limit, i.e., a voter may approve a set of projects which could not be realized within the given budget limit (in contrast to, e.g., Goel et al. [8]). Regarding the aggregation of the approval ballots, we follow the approach of Talmon and Faliszewski [17]. As a first step, we define a satisfaction function that returns for each voter and each possible committee the satisfaction of said voter based on her approval ballot. Then, an (ir)resolute budgeting method chooses a (set of) winning projects. One method is to output bundles that maximize the sum of the voters' satisfaction while taking into account the budget limit. As we will see more detailed in Section 3, this may lead to winner determination problems of high complexity in some cases. A different approach commonly used in practice is a simple greedy approach. In each iteration, the set of winning projects is extended by the project that maximizes the satisfaction in each incremental step, again respecting the budget limit. In this case, winner determination is more straightforward but depends on some tie-breaking mechanism in every step. Combinations of different satisfaction functions and budgeting methods have been studied by Talmon and Faliszewski [17] with respect to their axiomatic properties, see Baumeister et al. [4] for an adaption to irresolute variants of these rules. Unfortunately, many of the desired axioms are not satisfied by the proposed methods, which opens the possibility of manipulative interference on participatory budgeting processes.

Due to the combinatorial structure in participatory budgeting (i.e., the set of implemented items may not exceed the available funds), there are new types of control to consider. The axioms proposed by Talmon and Faliszewski [17] focus on the way a budgeting method should react to certain changes of the parameters. If, for example, an item's costs are less than originally anticipated, this should not lead to the item becoming unfunded. This is a reasonable assumption, however budgeting methods using a cost-based satisfaction function do not satisfy it. A budgeting method violating this axiom could be vulnerable to control if a chair would be able to influence an item's cost, in order to either exclude it from the chosen budget or to ensure it being funded. Another axiom requests that an increase of the budget limit may not lead to an item being excluded from the winning budget. As this axiom is not satisfied by any of the budgeting methods in question, this leaves the possible vulnerability to control via a change of the budget limit in order to in- or exclude an item. In order to examine these possible vulnerabilities to control further, we initially investigate the computational complexity of determining a winner for the greedy and maximizing rule in Section 3. Then we provide a general definition of manipulative interference in Section 4 with a specific focus on problems where either the budget limit or the cost of a specific item may be manipulated.

Related Work. Regarding traditional election problems this refers to different variants of control. Here, an election chair alters the structure of the election to make some distinguished candidate win or to prevent some distinguished candidate from winning. There is a huge amount of literature studying different kinds of control problems in voting. For an overview, we refer to the book chapter by Faliszewski and Rothe [6]. Related work on participatory budgeting close to our assumptions (i.e., approval ballots, binary outcomes, and satisfaction functions) was studied by Jain et al. [9], who considered satisfaction functions under project interactions, and Rey et al. [13], who embedded the framework introduced by Talmon and Faliszewski [17] into the framework of judgment aggregation. Aziz et al. [2] considered aggregation using an axiomatic approach instead of predefined rules. A well studied special case of approval-based participatory budgeting are multiwinner elections, where we assume uniform cost for each candidate. Lackner and Skowron [12] compare a variety of rules, that also use approval-based satisfaction functions as a measure of the voters' agreement with a committee.

2 Preliminaries

For a formal study of the voting step in participatory budgeting we follow the approach of Talmon and Faliszewski [17].

Definition 1. *A* ***budgeting scenario*** $E = (A, V, c, \ell)$ *consists of a set* $A = \{a_1, \dots, a_m\}$ *of* m *items, associated with a cost function* $c : A \to \mathbb{N}_+$*, and a set* $V = \{v_1, \dots, v_n\}$ *of* n *voters, where each voter* $v \in V$ *has an associated ballot* $A_v \subseteq A$ *containing a set of preferred items, and a budget limit* $\ell \in \mathbb{N}_+$.

Without giving a formal definition, let $\mathcal{E}$ denote the set of all possible budgeting scenarios without fixing any of the parameters (apart from mentioned dependencies of parameters in the definition above). The goal in participatory budgeting is to select a subset B of the items, called budget, such that the total cost of B does not exceed the budget limit ℓ. Slightly abusing notation we write $c(B) = \sum_{a \in B} c(a)$ to denote the total cost of some budget $B \subseteq A$. Moreover, we call a budget feasible if $c(B) \leq \ell$ and denote the set of feasible budgets by $\mathcal{B}(E) = \{B \subseteq A \mid c(B) \leq \ell\}$. Feasibility is a hard constraint, but of course the budget should take the ballots of the voters into account. Therefore, we introduce satisfaction functions for the voters.

Definition 2. *The satisfaction of a voter* $v \in V$ *with a given budget* $B \subseteq A$ *is modeled by a* ***satisfaction function*** $s : 2^A \times 2^A \to \mathbb{N}_0$*. For simplicity, we define* $B_v = A_v \cap B$ *to be the set of items, which are both, approved by a voter* v *and in a given budget* B*. In this paper we consider the following satisfaction functions focussing on:*

- ***quantity:*** *$s(A_v, B) = |B_v|$, the number of budgeted approved items, and*
- ***cost:*** *$s(A_v, B) = c(B_v)$, sum of the cost of the budgeted approved items.*

Slightly abusing notation, we write $s(V, B) = \sum_{v \in V} s(A_v, B)$ to denote the overall satisfaction of the voters in V with a budget B. The presented satisfaction functions follow different intentions and model different application scenarios. The intuition for satisfaction by quantity is straightforward. The satisfaction of a voter correlates with the number of implemented projects she likes. For satisfaction by cost, we assume satisfaction correlates with the amount of funds that are spent on preferred projects. Now, in order to compute a set of winning budgets based on the voters' preferences, we define an irresolute budgeting method $\mathcal{R}$, which maps a budgeting scenario E to a set of feasible budgets. The rules we study use the underlying satisfaction functions we defined previously.

Definition 3. *Given a budgeting scenario $E = (A, V, c, \ell) \in \mathcal{E}$ and a satisfaction function s we define:*

- ***Max rules (m):*** *as $\mathcal{R}^m_s(E) = \arg\max_{B \in \mathcal{B}(E)} s(V, B)$, and*
- ***Greedy rules (g):*** *starting with $B = \emptyset$ iteratively extend B by $a \in A \setminus B$, maximizing $s(V, B \cup \{a\})$, until there is no item $a \in A \setminus B$ with $c(B \cup \{a\}) \leq \ell$. Finally, set $\mathcal{R}^g_s(E) = \{B\}$.*

The max rules return all budgets that maximize the sum of the voters' satisfaction according to the function s. This rule is irresolute since there may be several budgets satisfying this requirement. In contrast, the greedy rules work iteratively. In each step one item that maximizes the sum of the voters' satisfaction when added to the current budget, will be added. We assume that some tie-breaking mechanism is used in each round, such that exactly one item is added. This leads to a resolute rule, always returning a set containing a single budget, also referred to as the budget returned by the rule. Together with the two satisfaction functions defined above, we consider four different rules.

Example 1. Let $E = (A, V, c, \ell)$ be a budgeting scenario with $A = \{a_1, a_2, a_3, a_4\}$, $V = \{v, v'\}$ with $A_v = A$ and $A_{v'} = \{a_1\}$, $c(a_i) = i$, and $\ell = 7$. For the greedy rules we break ties in favor of the item with a higher index. We have $\mathcal{R}^m_{|B_v|}(E) = \{\{a_1, a_2, a_3\}, \{a_1, a_2, a_4\}\}$, as both bundles yield a satisfaction of four, while the only bundle with a higher satisfaction is $A \notin \mathcal{B}(E)$. Similarly, it holds that $\mathcal{R}^m_{c(B_v)}(E) = \{\{a_1, a_2, a_4\}\}$ with a satisfaction of eight. For the greedy rules we list the items in the order they are added, that is $\mathcal{R}^g_{|B_v|} = \{\{a_1, a_4, a_2\}\}$ (where a_3 is skipped in the third iteration due to feasibility), and $\mathcal{R}^g_{c(B_v)} = \{\{a_4, a_3\}\}$.

In Section 4, we will define different decision problems related to different kinds of manipulative interference and study them from a computational point of view. As an intermediate step, it is important to determine the complexity for winner-determination problems first. Of course, computing a winning bundle for the greedy rule is easy, as it is a rather simple algorithm, that tries to find a solution that is close to the one from the max rule. Yet, in the following section, we will see that depending on the satisfaction function, there is little hope for an efficient algorithm that returns at least one budget that maximizes the sum of the voters' satisfaction. Our results range from polynomial-time algorithms to completeness in the polynomial hierarchy. We refer the reader to the textbook by Arora and Barak [1] for further details on computational complexity. In the rest of the paper, we assume that the reader is familiar with the complexity classes NP, coNP, $\Delta_2^p = \mathrm{P}^{\mathrm{NP}}$, and $\Sigma_2^p = \mathrm{NP}^{\mathrm{NP}}$. Further, in this paper, for a decision problem X, let $\overline{\mathrm{X}}$ denote its complement, and for $i, j \in \mathbb{N}_+$ with $i < j$ we denote $[i, j] = \{i, i+1, \ldots, j\}$, $[i, i] = \{i\}$, $[j, i] = \emptyset$, and $[i] = [1, i]$.

3 Winner Determination

In this section, we investigate the computational complexity for a variety of winner-determination problems associated with the considered budgeting rules. We assume, that for any greedy rule, a tie-breaking is fixed priorly and applied every round, resulting in a single final budget. Assuming that the given satisfaction function s and the tie-breaking rule are efficiently computable, computing a winning budget for a greedy rule can be done in polynomial time, since in each round the number of possible budgets that has to be considered equals the number of actually non-funded items. Therefore, we study decision problems related to winner determination only for maximizing rules combined with some efficiently computable satisfaction function s. The first problem we study asks whether there is some feasible budget where the sum of the voters' satisfaction exceeds some given bound. Additionally, we focus on some desired budget B^*, and ask whether it is a winning budget.

$\mathcal{R}_s$-Budget Score ($\mathcal{R}_s$-SC)	
Given:	A budgeting scenario $E = (A, V, c, \ell) \in \mathcal{E}$ and some bound $t \in \mathbb{N}_0$.
Question:	Is there a budget $B \in \mathcal{B}(E)$ with $s(V, B) \geq t$?

$\mathcal{R}_s$-Winning Budget ($\mathcal{R}_s$-WB)	
Given:	A budgeting scenario $E = (A, V, c, \ell) \in \mathcal{E}$ and some desired budget $B^* \subseteq A$.
Question:	Is $B^* \in \mathcal{R}_s(E)$?

Since the max rule we consider is irresolute, we also ask whether a given bundle is a subset of at least one, respectively every, winning budget. Formally the problem $\mathcal{R}_s^m$-Possibly Budgeted ($\mathcal{R}_s^m$-PB) has the same input as $\mathcal{R}_s^m$-WB, but the question is whether there is some $B \in \mathcal{R}_s^m(E)$ with $B^* \subseteq B$. Accordingly, we ask for the problem $\mathcal{R}_s^m$-Necessarily Budgeted ($\mathcal{R}_s^m$-NB), whether $B^* \subseteq B$ for every budget $B \in \mathcal{R}_s^m(E)$. Now, we provide general upper bounds.

Lemma 1. *Consider $E \in \mathcal{E}$, an efficiently computable satisfaction function s, and $t^* = \max_{B \in \mathcal{B}(E)} s(V, B)$. For $\mathcal{R}_s^m$-SC being a member of complexity class $\mathcal{A}$, it holds that*

(i) $\mathcal{R}_s^m$-WB *is in* co$\mathcal{A}$*, and*
(ii) $\mathcal{R}_s^m$-PB *and* $\mathcal{R}_s^m$-NB *are in* $\mathrm{P}^{\mathcal{A}[O(\log(t^*))]}$*, and*
(iii) $\mathcal{R}_s^m$-SC $\in$ NP.

Proof. Consider a budgeting scenario E, a set of items $B^* \subseteq A$, and a satisfaction function s. For (i) we first may verify if B^* is feasible and compute $s(V, B^*)$ in polynomial time. Then to solve $\mathcal{R}_s^m$-WB we may decide in co$\mathcal{A}$ if every feasible budget has an overall satisfaction of less than $s(V, B^*) + 1$ by solving $\overline{\mathcal{R}_s^m\text{-SC}}$.

For (ii) we may compute the optimal score t^* of a winning budget by sending $O(\log(t^*))$ queries to an $\mathcal{A}$-oracle using binary search. To solve $(E, B^*) \in \mathcal{R}_s^m$-PB we construct another satisfaction function s' with $s'(V, B) = 2 \cdot s(V, B) + 1$ if $B^* \subseteq B$ and $s'(V, B) = 2 \cdot s(V, B)$ otherwise. To answer $(E, B^*) \in \mathcal{R}_s^m$-PB, we send a final query to our $\mathcal{A}$-oracle, asking whether $(E, 2t^* + 1) \in \mathcal{R}_{s'}$-SC is a yes-instance. We can use similar techniques to solve $(E, B^*) \in \mathcal{R}_s^m$-NB, by defining s', such that a bundle B with $B^* \nsubseteq B$ is assigned the slightly increased score. Then $(E, B^*) \in \mathcal{R}_s^m$-NB is a yes-instance if and only if $(E, 2t^* + 1) \in \mathcal{R}_{s'}$-SC is a no-instance. Overall we can query an $\mathcal{A}$-oracle $O(\log(t^*))$ times.

For (iii) recall that s is efficiently computable, so verifying that there is a budget B with $s(V, B) \geq t$ can be done in polynomial time. □

From the above lemma it follows that if $\mathcal{R}_s^m$-SC is efficiently computable for some satisfaction function s, then the other winner-determination problems are also in P. Another implication is, that $\mathcal{R}_s^m$-PB and $\mathcal{R}_s^m$-NB are in Δ_2^p in general, and in $\Theta_2^p = \mathrm{P}^{\mathrm{NP}[\log]}$ for satisfaction functions, where the satisfaction for a bundle is at most polynomial in the (binary encoded) size of the budgeting scenario. For the rules we consider, we will see that P and Δ_2^p are suitable upper bounds for $\mathcal{R}_s^m$-PB and $\mathcal{R}_s^m$-NB. Yet, there are satisfaction functions,[1] for which $\mathcal{R}_s$-SC is known to be NP-complete, but for $E = (A, V, c, \ell)$ and $B \subseteq A$, the score $s(V, B)$ is bounded by $|A| \cdot |V|$, yielding an upper bound of Θ_2^p (which is not necessarily tight).

We continue by establishing tight bounds for the winner determination problems. Talmon and Faliszewski [17] already showed, that $\mathcal{R}_{|B_v|}^m$-SC is solvable in polynomial time. Following Lemma 1 we can formulate the following corollary.

Corollary 1. $\mathcal{R}_{|B_v|}^m$-WB*,* $\mathcal{R}_{|B_v|}^m$-PB*, and* $\mathcal{R}_{|B_v|}^m$-NB *are in* P*.*

Next, we establish lower bounds for $\mathcal{R}_{c(B_v)}^m$. Talmon and Faliszewski [17] showed NP-hardness for $\mathcal{R}_{c(B_v)}^m$-SC by reducing from the well known problem Subset Sum (see Garey and Johnson [7]).

Theorem 1. $\mathcal{R}_{c(B_v)}^m$-WB *is* coNP*-complete, and* $\mathcal{R}_{c(B_v)}^m$-PB *and* $\mathcal{R}_{c(B_v)}^m$-NB *are* Δ_2^p*-complete.*

Proof. We start by showing coNP-hardness for $\mathcal{R}_{c(B_v)}^m$-WB. We will reduce from $\overline{\text{Subset Sum}}$, where the input is a set of integers $N = \{n_1, \ldots, n_m\} \subseteq \mathbb{N}_+$

[1] For example the Chamberlin-Courant rule for approval ballots, studied by Skowron and Faliszewski [16].

and a bound $n \in \mathbb{N}_+$, and the question is, whether there is no subset $S \subseteq N$ with $\sum_{i\in S} i = n$. We transform an arbitrary instance (N, n) to an instance of $\mathcal{R}^m_{c(B_v)}$-WB. Let $A = \{a_1, \ldots, a_m, b\}$, $V = \{v\}$ with $A_v = A$, $c(a_i) = 2n_i$ for each $i \in [1, m]$, and $c(b) = 2n - 1$, and $\ell = 2n$. Finally, we set $B^* = \{b\}$ and claim that $(E, B^*) \in \mathcal{R}^m_{c(B_v)}$-WB if and only if $(N, n) \in \overline{\text{SUBSET SUM}}$. In particular, B^* is a winning budget if there is no set of items, which adds up to a cost of $2n$. This is exactly the case if there is no $S \subseteq N$ which sums up to n.

To show Δ^p_2-completeness for the remaining problems, we will use the following Δ^p_2-complete problem, based on Krentel's results [11, Thm 2.1, Thm 3.3].[2]

EVEN SUBSETSUM (ESS)	
Given:	A finite set of integers $N \subset \mathbb{N}_+$ and a distinct integer $n \in \mathbb{N}_+$.
Question:	Let $t = \sum_{i\in S} i$ be the largest possible value with $t \leq n$ over all $S \subseteq N$. Is $t \mod 2 \equiv 0$?

For an upper bound, $\mathcal{R}^m_{c(B_v)}$-PB and $\mathcal{R}^m_{c(B_v)}$-NB are in Δ^p_2 following Lemma 1. Since the overall satisfaction derived from a winning budget t^* depends on the cost, $\log(t^*)$ is polynomial in the instance size (but not logarithmical).

To show hardness, we reduce from ESS. Consider any ESS instance $\mathcal{I} = (N, n)$ with $N = \{n_1, \ldots, n_m\}$. For simplicity and without loss of generality assume that $n \geq n_i$ holds for every $i \in [m]$. Further, let $k = |n|$ denote the length of the binary representation of n. We construct a $\mathcal{R}^m_{c(B_v)}$-PB instance $\mathcal{I}' = (E, B^*)$, with $A = \{a_1, \ldots, a_m, b_1, \ldots, b_k\}$ and $V = \{v\}$ with $A_v = A$. For our cost function c, we interpret the cost $c(a)$ of some item $a \in A$ in its binary encoding. By construction, each cost $c(a)$ will be consisting of two different zones, which are k bit long (to prevent carries). The front zone will be used to verify if the maximum achievable cost is even and the end zone will be used to still respect our bound n. For each a_i we will simply set cost n_i in both zones, i.e., $c(a_i) = (2^{k+1} + 1) \cdot n_i$. For each b_i we will only use the front zone to set the i-th bit to one, i.e., $c(b_i) = 2^{i+k}$. We choose ℓ in a way that the first k bits are set to one and the last k bits are set to the binary representation of n, that is $\ell = 2^{k+1}(2^{k+1} - 1) + n$. Finally, we set $B^* = \{b_1\}$.

To prove equivalence, note that each winning budget in $\mathcal{I}'$ has a satisfaction of at least $\sum_{i=1}^{k} c(b_i) = 2^{k+1}(2^{k+1} - 1)$ (e.g., by budgeting all b_i) and at most ℓ (by definition). Also note, that the cost of any budget not containing any b_i, n is always either exceeded in both or neither zones simultaneously. Therefore, any winning budget in $\mathcal{I}'$ has an equivalent cost in the last zone to the largest possible value for $\mathcal{I}$, while the front zone can always be filled up bitwise by values b_i. Finally, note, that by construction b_1 is only part of a winning budget if and only if the optimal value adds up to an even value, so its corresponding bit in the front zone can be flipped to one by adding b_1. By construction $B^* = \{b_1\}$ is necessarily and thus possibly budgeted if and only if $\mathcal{I}$ is a yes instance. □

[2] Also known as a KNAPSACK variant in related literature (see Kellerer et al. [10]).

4 Manipulative Interference

To study problems of manipulative interference in a generic way, we consider an alteration function f, that maps from a given budgeting scenario to a set of possible scenarios after the alteration of specified parameters (e.g., the cost function or the voters' ballots). Formally, we have $f : \mathcal{E} \to 2^{\mathcal{E}}$. We assume, that we can efficiently verify whether $E' \in f(E)$ holds. We distinguish between a constructive and a destructive variant of manipulative interference.

Constructive-$\mathcal{R}_s$-Manipulative-Interference (C-$\mathcal{R}_s$-MI)	
Given:	A budgeting scenario E, a set of items $B_\heartsuit$, an integer k, and an alteration function f.
Question:	Is there a budgeting scenario $E' \in f(E)$, such that there is a winning budget $B \in \mathcal{R}_s(E')$ with $\lvert B_\heartsuit \cap B\rvert \geq k$?

For Destructive-$\mathcal{R}_s$-Manipulative-Interference (D-$\mathcal{R}_s$-MI) the input remains the same, but now we ask whether there is a budgeting scenario $E' \in f(E)$, such that there is a winning budget $B \in \mathcal{R}_s(E')$ with $\lvert B_\heartsuit \cap B\rvert < k$. Both definitions are very general. In particular, we have a set $B_\heartsuit$ of distinguished items. A natural restriction is the focus on a single item with $\lvert B_\heartsuit\rvert = 1$. In the constructive case we ask, whether there is a winning budget that contains at least k of the preferred items. This again gives the freedom to choose between having at least one to having all items in the winning budget. Accordingly, in the destructive case we ask whether there is a winning budget where less than k of the distinguished items are included. By setting $k = 1$ we obtain the special case where we ask for a winning budget containing none of the items in $B_\heartsuit$. A more strict variant of constructive manipulative interference would be to require that all winning budgets contain at least k of the preferred items. Accordingly, in the destructive variant one could require that the condition holds for all winning budgets. In this paper, we will however focus on the above presented variants.

For a trivial upper bound, we may guess an altered budgeting scenario $E' \in f(E)$ and a budget $B \in \mathcal{B}(E')$ with $\lvert B_\heartsuit \cap B\rvert \geq k$, and verify whether $B \in \mathcal{R}_s(E')$ holds by querying an oracle to answer $(E', B) \in \mathcal{R}_s$-WB.

Lemma 2. *Fix some alteration function f such that $\mathcal{R}_s$-WB restricted to budgeting scenarios E' with $E' \in f(E)$ is in $\mathcal{A}$. Then*

(i) C-$\mathcal{R}_s$-MI and D-$\mathcal{R}_s$-MI restricted to f are in $\mathrm{NP}^{\mathcal{A}}$, and
(ii) $\mathcal{R}_s$-WB restricted to f is in $\mathcal{A}$.

Hence, any form of manipulative interference, like manipulation, bribery, or control in classical voting, is bound upwards by NP, for rules, where $\mathcal{R}_s$-WB can be solved efficiently, including all greedy rules. Following Lemma 1, an upper bound for all maximizing rules is Σ_2^p. For lower bounds, we investigate specific forms of control, as a subtype of manipulative interference, to determine how vulnerable the rules in question are to seemingly small changes of a given budgeting scenario. In particular, we study the impact of influencing the budget limit or an item's cost on the outcome. While initially putting their combinatorial budgeting methods forward, Talmon and Faliszewski [17] simultaneously proposed several axioms a budgeting method should satisfy. As these axioms are not satisfied by, in some cases any and in other cases several of the proposed rules, we derive ways in which to exploit these particular weaknesses in order to exert control over the results of the participatory budgeting process. We investigate tight bounds for these specific forms of control, by studying the complexity of $\mathcal{R}_s$-MI under respective alteration functions f. Table 1 summarizes our results.

Changing the Budget Limit. The first type of control we consider is by altering the budget limit, which originates from the axiom of limit monotonicity as defined by Talmon and Faliszewski [17]. The idea is that if the budget limit is increased, no previously budgeted item becomes unfunded. All budgeting rules we consider violate said axiom. Thus, we define a variant of manipulative interference capturing different possibilities of taking influence on the budget limit.

Definition 4. *Given* $L, H \in \mathbb{N}_+$ *with* $L \leq H$*, define an alteration function* $f_{L,H}$ *such that* $(A, V, c, d) \in f_{L,H}(E)$ *for every* $E = (A, V, c, \ell)$ *and* $d \in [L, H]$*. The restriction of manipulative interference to such alteration functions and* $k \leq |B_\heartsuit|$ *will be denoted by* $\mathcal{R}_s$-Control-by-Setting-the-Budget-Limit *(*$\mathcal{R}_s$-CSBL*).*

In the constructive case C-$\mathcal{R}_s$-CSBL asks whether it is possible to increase or decrease the budget limit such that at least k of the desired items are contained in one winning budget. In the destructive variant D-$\mathcal{R}_s$-CSBL, asks whether it is possible to obtain a winning budget containing less than k of the distinguished items by increasing or decreasing the budget limit. Since the rules we consider here violate limit monotonicity, they are obviously vulnerable to this type of control. Now, we will show that for the max rules and the quantity based satisfaction functions both control problems are solvable in polynomial time.

Theorem 2. C-$\mathcal{R}^m_{|B_v|}$-CSBL *and* D-$\mathcal{R}^m_{|B_v|}$-CSBL *are in* P.

Proof. We start with the constructive variant, showing C-$\mathcal{R}^m_{|B_v|}$-CSBL $\in$ P. Let $B_\heartsuit \subseteq A$ be the set of items, from which we want to include at least k items, by a successful control, in at least one winning budget. We reduce the given instance $\mathcal{I} = (E, B_\heartsuit, k, f_{L,H})$ to $\mathcal{I}' = (E', B_\heartsuit, k, f_{L,H})$, by modifying the set of voters. For $w = |B_\heartsuit|$, we clone each voter $w+1$ times and add one additional voter v with $A_v = B_\heartsuit$, resulting in a set of voters V'. This enforces, that budgets containing more items from $B_\heartsuit$ yield a slightly higher satisfaction in case of ties. We set $E' = (A, V', c, \ell)$. It holds that $\mathcal{I} \in$ C-$\mathcal{R}^m_{|B_v|}$-CSBL $\Leftrightarrow$ $\mathcal{I}' \in$ C-$\mathcal{R}^m_{|B_v|}$-CSBL, because for every $d \in [L, H]$ and any two budgets $B \in \mathcal{R}^m_{|B_v|}((A, V, c, d))$ and $B' \in \mathcal{R}^m_{|B_v|}((A, V', c, d))$ it holds that $|B \cap B_\heartsuit| \leq |B' \cap B_\heartsuit|$. Note, that for E' the maximum achievable satisfaction for any feasible budget in $\mathcal{B}(E')$ is at most $s(V', A) = (w + 1) \cdot s(V, A) + w$. We use dynamic programming as described by Talmon and Faliszewski [17], to determine the minimum cost of a budget with a satisfaction of exactly t for each $t \in [0, s(V', A)]$, for the budgeting scenario E'. We may compute those values and store them in a list $\mathcal{T}$. Formally, for every $t \in [0, s(V', A)]$, if there is no feasible budget with a satisfaction of exactly t, let $\mathcal{T}(t) = \infty$ and otherwise, let $\mathcal{T}(t) = \min_{B' \in \{B \in \mathcal{B}(E') | s(V',B)=t\}} c(B)$. Finally, we solve C-$\mathcal{R}^m_{|B_v|}$-CSBL by identifying if there is a value $d \in [L, H]$ we can set the budget limit to, such that there is a winning budget B with $|B \cap B_\heartsuit| \geq k$. We can search for d in a polynomial number of steps. First, we initialize to $d = H$ and determine the highest value t^* with $\mathcal{T}(t^*) \leq d$. We express this value as $t^* = (w+1) \cdot t_1 + t_2$, such that $t_2 \in [0, w]$. If $t_2 \geq k$, a control can be executed by choosing $\ell = d$. Otherwise, we can decrease d to $d = \mathcal{T}(t^*) - 1$ and repeat until we either found d, or stop if $d < L$. Note, that this procedure stops after at most $s(V', A) < |A| \cdot |V'|$ steps.

To show, that D-$\mathcal{R}^m_{|B_v|}$-CSBL $\in$ P also holds, we can use the same algorithm. We deviate by slightly permutating the values for the function $\mathcal{T}$, such that bundles are preferred, that include less items from $B_\heartsuit$. In particular, for every $t_1 \in [0, w+1]$ and $t_2 \in [0, w]$, we set $\mathcal{T}'((w+1) \cdot t_1 + t_2) = \mathcal{T}((w+1) \cdot t_1 + w - t_2)$. Again, we search for $d \in [L, H]$, starting at $d = H$, while our condition for identifying a yes-instance changes to $t_2 \geq w - k$. □

For C-$\mathcal{R}^m_{c(B_v)}$-CSBL and D-$\mathcal{R}^m_{c(B_v)}$-CSBL tight complexity bounds are still open. Following Lemma 2, both problems are in Σ_2^p and following Theorem 1 they are Δ_2^p hard. The latter follows easily, as problems of winner-determination can be reduced to control problems without altering the parameters at all.

A very general result holds for all additive satisfaction functions, i.e., for any function s with $s(A_v, B) = \sum_{a \in A_v} \sum_{b \in B} s(\{a\}, \{b\})$ for all $A_v, B \subseteq A$.

Theorem 3. *For additive satisfaction functions s it holds that* C-$\mathcal{R}^g_s$-CSBL *and* D-$\mathcal{R}^g_s$-CSBL *are in* P.

Proof. Since s is additive by assumption and not dependent on the budget limit $d \in [L, H]$, the processing order of a greedy rule $\mathcal{R}^g_s$ is determined prior execution, using a fixed linear tie-breaking scheme $\succ$ if necessary. Without loss of generality we assume, the set of items is labeled in this ordering. That is, for $A = \{a_1, \ldots, a_m\}$ we assume that for each $1 \leq i < j \leq m$ it holds that either $s(V, \{a_i\}) > s(V, \{a_j\})$ or $s(V, \{a_i\}) = s(V, \{a_j\})$ and $a_i \succ a_j$. Further, we denote $A_i = \{a_1, \ldots, a_i\}$ and $E_d = (A, V, c, d)$.

We use dynamic programming to compute all values for $d \in [L, H]$, such that we can include exactly $j \in [0, |B_\heartsuit|]$ items from $B_\heartsuit$, only using items from A_i for $i \in [0, m]$. We generate a $(|B_\heartsuit| + 1) \times (m + 1)$ table $\mathcal{T}$, where each column represents a processing step after investigating an item a_i and each row represents the number of items shared with $B_\heartsuit$ in a possible (partial) solution. More precisely, the leftmost column ($i = 0$) represents an initial state, column i represents partial solutions after processing the first i items A_i, and the values in the rightmost column ($i = m$) represent possible (full) solutions.

The intuition behind $\mathcal{T}(j, i)$ is, that the greedy rule has already executed its first i iterations for an unknown budget limit $d \in [L, H]$, such that j items from $B_\heartsuit$ have already been added to the (partial) budget B_i. As we might have added in items in some iterations, we assume that some of the budget has already been filled by the respective items cost $c(B_i)$. For $d \in [L, H]$ and every possible resulting partial budget $B_i = \mathcal{R}_s^g(E_d) \cap A_i$ containing j items from $B_\heartsuit$, we add $d - c(B_i) \in \mathcal{T}(j, i)$. Note that $\mathcal{T}(j, i)$ is empty, if there is no $d \in [L, H]$ such that $\mathcal{R}_s^g(E_d)$ contains exactly j items from $B_\heartsuit$ after the first i iterations, i.e., if there is no $d \in [L, H]$ with $|\mathcal{R}_s^g(E_d) \cap A_i \cap B_\heartsuit| = j$. In particular, $\mathcal{T}(j, i)$ contains every value, such that we can extend the cost $c(B_i)$ of a partial budget $B_i = \mathcal{R}_s^g(E_d)$ satisfying above conditions to retrieve the input value d. Additionally, we claim that each $\mathcal{T}(j, i)$ can be represented by two discrete intervals, such that we can encode the values for each cell efficiently (to be shown at the end of the proof).

We initialize every cell to $\mathcal{T}(j, i) = \emptyset$ for $j \in [0, |B_\heartsuit|]$ and $i \in [0, m]$, except for $\mathcal{T}(0, 0) = [L, H]$. Next, we populate $\mathcal{T}$ left-to-right and top-to-bottom, where any cell $\mathcal{T}(j, i)$ is used to extend $\mathcal{T}(j, i+1)$ and $\mathcal{T}(j+1, i+1)$, i.e., we only populate to the right. By design, each cell might be populated from two different cells; in this case we consider the union of both values. We will explain in detail how to populate in the first iteration ($i = 1$) to generalize from there.

We start by investigating $\mathcal{T}(0, 0)$ and reduce our problem to smaller instances, where the decision on a_1 is already made and thus, we only need to consider $A \setminus A_1$ in following iterations. In particular, we study two main cases. In case $d < c(a_1)$ holds, then in the first iteration we cannot add a_1 to the bundle. Hence, in case $d \in [L, c(a_1) - 1]$, we can reduce to an instance, which considers only $A \setminus A_1$, i.e., we extend cell $(0, 1)$ by $\mathcal{T}(0, 1) = \mathcal{T}(0, 1) \cup [L, c(a_1) - 1]$. Otherwise, for $d \geq c(a_1)$, we certainly need to add a_1 to the budget in this iteration. Again, we can reduce this to an instance not considering a_1, by choosing the budget limit, such that $d \geq c(a_1)$ holds in any case. Instead of enforcing d to have a minimum value (of at least $c(a_1)$), we reduce by decreasing the respective values to choose from by $c(a_1)$. In case $a_1 \notin B_\heartsuit$, we set $\mathcal{T}(0, 1) = \mathcal{T}(0, 1) \cup [0, H - c(a_1)]$, otherwise we also increment j, i.e., $\mathcal{T}(1, 1) = \mathcal{T}(1, 1) \cup [0, H - c(a_1)]$.

More general, for some iteration, in which we investigate the cell (j, i), we again study two seperate cases. We split $\mathcal{T}(j, i)$ into two disjoint sets based on the respective items cost $c(a_i)$. That is, $X = \mathcal{T}(j, i) \cap [0, c(a_i) - 1]$ and $Y = \mathcal{T}(j, i) \cap [c(a_i), H]$. We extend $\mathcal{T}(j, i+1)$ by X. Before extending a cell with values from Y, we shift all values of Y by $-c(a_i)$. Formally, that is $Y' = \{y - c(a_i) \mid y \in Y\}$. Finally, if $a_i \notin B_\heartsuit$, we extend $\mathcal{T}(j, i+1)$ by Y', otherwise we extend $\mathcal{T}(j+1, i+1)$ by Y'.

After populating the table $\mathcal{T}$, there is a $d \in [L, H]$ with $|\mathcal{R}_s^g(E_d) \cap B_\heartsuit| = j$ if and only if $\mathcal{T}(j, m) \neq \emptyset$. Additionally, we can use backtracking on every value $d' \in \mathcal{T}(j, m)$, to compute a distinct value $d \in [L, H]$ with $|\mathcal{R}_s^g(E_d) \cap B_\heartsuit| = j$.

It is left to show, that each cell of the table can be stored efficiently. Therefore, we show, that each cell can be represented with at most two intervals $I_1, I_2 \subseteq [0, H]$ with $0 \in I_1$. Of course, this claim holds for $\mathcal{T}(0, 0) = [L, H]$ by assumption and for the remaining values in the leftmost column, as they are never populated. Next, we show that, the if the claim holds for previously populated cells, then it also holds after populating the next cell. We start with the first row. Consider some iteration, where we are investigating cell $\mathcal{T}(0, i)$. For simplicity, we imagine the (at most) two intervals in $\mathcal{T}(0,i)$ to occupy respective space on the larger interval $[0,H]$. We imagine this interval to be ordered left-to-right by ascending values. In any iteration investigating $\mathcal{T}(j,i) \subseteq [0,H]$, we split $\mathcal{T}(j,i)$ at $c(a_i)$. The left part (excluding $c(a_i)$) is added to $\mathcal{T}(0,i+1)$ without any shifting operation. If $a_i \in B_\heartsuit$, we use the values on the right (including $c(a_i)$) to populate $\mathcal{T}(1,i+1)$, which is not in the first row. Otherwise, we shift those values to the left by subtracting $c(a_i)$ and add them to $\mathcal{T}(0,i+1)$. If $c(a_i)$ did not intersect one of the intervals, the claim holds. If on the other hand $c(a_i)$ did intersect an interval, then the rightmost part is shifted to the left, such that the starting value is 0. By assumption in $\mathcal{T}(0,i+1)$ there are now two intervals starting with 0. Thus, those two intervals collapse to a single interval. For the remaining rows first note, that if we split and shift any interval $[0,x]$, the two resulting intervals both have a starting value of 0. Subsequently, the only way there is an interval $I \in \mathcal{T}(j,i)$ with $j > 0$ and $0 \notin I$, is that in some previous iteration i' a preferred item $a_{i'}$ was added, whose cost $c(a_{i'})$ did not intersect the right interval in $\mathcal{T}(j-1,i')$. In particular, the right interval was shifted to the left and added to $\mathcal{T}(j,i'+1)$. This especially means, that $\mathcal{T}(j-1,i'+1)$ can only hold the left interval, which is always sticking to 0 when using the operations of splitting and shifting. Overall, in each column there can be at most one interval I with $0 \notin I$. □

Changing an Item's Cost. Another type of control is the alteration of a given item's cost. This is based on the axiom of discount monotonicity, introduced by Talmon and Faliszewski [17]. The intuition is that decreasing the cost of a budgeted item does not lead to it being not funded anymore. Using a budgeting method that satisfies this axiom means that there is no incentive to strategize regarding an item's price. Otherwise, one might not take an offer that would reduce the cost of an item, fearing that it could lead to eliminating that item from the winning bundle. This is not desirable, as it would be a waste of resources.

Definition 5. *Given $a_\heartsuit \in A$ and $L, H \in \mathbb{N}_+$ with $L \leq H$, define an alteration function $f_{L,H}$ with $(A, V, c', \ell) \in f_{L,H}(E)$ for every $E = (A, V, c, \ell)$ and $d \in [L, H]$ such that $c'(a_\heartsuit) = d$ and $c'(a) = c(a)$ for all $a \in A \setminus \{a_\heartsuit\}$. The restriction of manipulative interference to such alteration functions, $B_\heartsuit = \{a_\heartsuit\}$, and $k = 1$ will be denoted by $\mathcal{R}_s$-Control-by-Setting-an-Item's-Cost ($\mathcal{R}_s$-CSIC).*

With the above defined restrictions, C-$\mathcal{R}_s$-CSIC asks whether the cost of the desired item $a_\heartsuit$ can be changed within the given bounds such that a winning budget contains $a_\heartsuit$. In D-$\mathcal{R}_s$-CSIC we ask whether we can obtain a winning budget that does not contain $a_\heartsuit$. The complexity of $\mathcal{R}_s$-CSIC in both variants for $\mathcal{R}^m_{|B_v|}$ and $\mathcal{R}^g_{|B_v|}$ follow directly from the results by Talmon and Faliszewski [17] and Baumeister et al. [4]. As both rules satisfy discount monotonicity, the strategy is to set $d = L$ for the constructive variant and $d = H$ for the destructive variant. To see if the control attempt was successful, we can solve the respective winner determination problems, which both are in P.

Corollary 2. C-$\mathcal{R}^m_{|B_v|}$-CSIC*,* D-$\mathcal{R}^m_{|B_v|}$-CSIC*,* C-$\mathcal{R}^g_{|B_v|}$-CSIC*, and* D-$\mathcal{R}^g_{|B_v|}$-CSIC *are in* P.

We turn to the cost satisfaction function and show that for the maximizing rule the constructive variant of setting an item's cost is complete for Δ_2^p.

Theorem 4. C-$\mathcal{R}^m_{c(B_v)}$-CSIC *is* Δ_2^p*-complete.*

Proof. For a lower bound, following Theorem 1, $\mathcal{R}^m_{c(B_v)}$-PB is Δ_2^p-complete. Subsequently, C-$\mathcal{R}^m_{c(B_v)}$-CSIC is at least Δ_2^p-hard, as it coincides with $\mathcal{R}^m_{c(B_v)}$-PB if we choose $f_{L,H}$ such that $L = H = c(a_\heartsuit)$ for the item $a_\heartsuit$ with $B_\heartsuit = \{a_\heartsuit\}$.

Next, we want to show a matching upper bound. Let $A' = A \setminus \{a_\heartsuit\}$, $E' = (A', V, c, \ell)$, and $E_d = (A, V, c', \ell)$ with $c'(a_\heartsuit) = d$ and $c'(a) = c(a)$ for all $a \in A'$. First we compute the overall satisfaction t^* of a winning budget for E', which can be done as described in the proof of Lemma 1 by querying an NP-oracle a polynomial number of times. Knowing the optimal score for a winning budget not containing $a_\heartsuit$, we can query an NP-oracle to solve C-$\mathcal{R}^m_{c(B_v)}$-CSIC. In particular, we ask whether there exists $d \in [L, H]$, such there exists $B \in \mathcal{B}(E_d)$ with $a_\heartsuit \in B$ and $s(V, B) \geq t^*$. Finding an answer to this question is in NP. The answer is yes, if and only if there exists a $d \in [L, H]$, such that there is a budget containing $a_\heartsuit$, that yields a satisfaction at least as high as any bundle not containing $a_\heartsuit$. □

Note that the above proof does not hold for the destructive control variant, although the lower bound holds for similar reasons. Knowing t^*, does not lead to a bounded number of obvious NP questions. Instead, we still need to determine, whether there exists a $d \in [L, H]$, such that every feasible bundle containing $a_\heartsuit$ yields a satisfaction of at most t^*. For the greedy rule and the cost satisfaction function we can again show polynomial-time solvability.

Theorem 5. C-$\mathcal{R}^g_{c(B_v)}$-CSIC *and* D-$\mathcal{R}^g_{c(B_v)}$-CSIC *are in* P.

Proof. Consider any budgeting scenario E and a given item $a_\heartsuit$, which should be included (or excluded) into the (resolute) final outcome. Further, let $E_d = (A, V, c', \ell)$ denote the modified budgeting scenario with $c'(a_\heartsuit) = d$ and $c'(a) = c(a)$ for every $a \in A \setminus \{a_\heartsuit\}$. We assume, that there is a linear tie-breaking scheme $\succ$ over the set of items A, which is identical for every E_d.

Note that $s(V, A) = \sum_{v \in V} \sum_{a \in B} c(a)$ is an additive function. Hence, the order in which the greedy rule $\mathcal{R}^g_{c(B_v)}$ determines, whether to add an item or not, is never changing during execution. Yet, the position of $a_\heartsuit$ in this order also depends on its cost $c(a_\heartsuit)$. Formally, let the position of $a \in A$ in the processing order with respect to E_d and $\succ$ be denoted by $\mathrm{pos}(a, E_d, \succ)$. To solve C-$\mathcal{R}^g_{c(B_v)}$-CSIC, we compute $\mathcal{R}^g_{c(B_v)}(E_d)$ for at most $|A|$ values $d \in [L, H]$. Initially we set $d = L$ and compute the winning budget. If $a_\heartsuit \in \mathcal{R}^g_s(E_d)$, the input is a yes-instance. Otherwise, we increase d to the minimum value d', such that $\mathrm{pos}(a_\heartsuit, E_{d'}, \succ) < \mathrm{pos}(a_\heartsuit, E_d, \succ)$. Precisely, for the item a with $\mathrm{pos}(a, E_d, \succ) = \mathrm{pos}(a_\heartsuit, E_d, \succ) - 1$ we set $d' = \left\lceil \sum_{v \in V} c(A_v \cap \{a\}) / \sum_{v \in V} |A_v \cap \{a_\heartsuit\}| \right\rceil$. If necessary due to a tie, which is broken favoring a, d' is additionally increased by 1. Again, if $a_\heartsuit \notin \mathcal{R}^g_s(E_{d'})$ holds, we relabel d' to d and repeat the last step, until we cannot increase the cost of item $a_\heartsuit$ without exceeding our upper limit H. If this is the case, we have successfully identified of a no-instance.

To solve D-$\mathcal{R}^{g}_{c(B_v)}$-CSIC, we use a similar technique. Now, we initialize $d = H$ and decrease d to the highest value, such that the decision, whether to add $a_{\heartsuit}$, is done one step later in the processing order. We hold and output yes, if for any such d' it holds that $a_{\heartsuit} \notin \mathcal{R}^{g}_{c(B_v)}(E_{d'})$, and output no, if d' falls below L. □

5 Conclusions

We extended the study of winner determination problems for the considered budgeting methods, and introduced a general form of manipulative interference. We focussed on two restrictions, the problems of setting the budgeting limit and setting an item's cost. The results are summarized in Table 1. For most of the rules the problems are solvable in P, whereas they are Δ_2^p-hard for the maximizing rule combined with the cost satisfaction function. This correlates with the results obtained for winner determination, where the associated decision problems are complete for coNP and Δ_2^p.

When studying problems of manipulative interference, polynomial-time algorithms are usually undesired, as this does not offer any protection. However, this can also be interpreted from the perspective of robustness. In reality, the budget limit and the cost of an item may both not be perfectly accurate, meaning that there may be some uncertainty about parts of the budget, or that the cost is rather an estimate. Then problems of manipulative interference give insight in how vulnerable the actual solution may be to changes in one of these parameters.

We considered two of the axioms studied by Talmon and Faliszewski [17]. As a task for future research, this should be extended to other axioms and other types of control that are specific for participatory budgeting. Due to our general formulation of manipulative interference, some of our results may still apply. Another task would be, to close the gap between upper and lower bound for the maximizing rule with the cost satisfaction function. The study can also be extended to other budgeting methods. For example, a satisfaction function could also yield dissatisfaction for rejected projects, or a voting rule could measure the overall satisfaction by the minimum voter's satisfaction instead of the sum.

Acknowledgment. We thank all reviewers for their helpful comments. Supported in part by DFG grant BA 6270/1-1 and by the project "Online Participation," funded by the NRW Ministry for Innovation, Science, and Research.

Table 1. Summary of our complexity results for respective control problems.

$\mathcal{R}_s$	Setting-the-Budget-Limit		Setting-an-Item's-Cost	
	Constructive	Destructive	Constructive	Destructive
$\mathcal{R}^{g}_{\lvert B_v\rvert}$	in P	in P	in P	in P
$\mathcal{R}^{g}_{c(B_v)}$	in P	in P	in P	in P
$\mathcal{R}^{m}_{\lvert B_v\rvert}$	in P	in P	in P	in P
$\mathcal{R}^{m}_{c(B_v)}$	Δ_2^p-h., in Σ_2^p	Δ_2^p-c.	Δ_2^p-h., in Σ_2^p	Δ_2^p-h., in Σ_2^p

References

1. Arora, S., Barak, B.: Computational Complexity: A Modern Approach. Cambridge University Press (2009)
2. Aziz, H., Lee, B.E., Talmon, N.: Proportionally Representative Participatory Budgeting: Axioms and Algorithms. In: Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems. pp. 23–31. IFAAMAS (2018)
3. Aziz, H., Shah, N.: Participatory Budgeting: Models and Approaches. In: Rudas, T., Gábor, P. (eds.) Pathways between Social Science and Computational Social Science: Theories, Methods and Interpretations, pp. 215–236. Springer (2021)
4. Baumeister, D., Boes, L., Seeger, T.: Irresolute Approval-based Budgeting. In: Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems. pp. 1774–1776. IFAAMAS (2020)
5. Cabannes, Y.: Participatory Budgeting: A Significant Contribution to Participatory Democracy. Environment and Urbanization **16**(1), 27–46 (2004)
6. Faliszewski, P., Rothe, J.: Control and Bribery in Voting. In: Brandt, F., Conitzer, V., Endriss, U., Lang, J., Procaccia, A. (eds.) Handbook of Computational Social Choice, chap. 7, pp. 146–168. Cambridge University Press (2016)
7. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman and Company (1979)
8. Goel, A., Krishnaswamy, A.K., Sakshuwong, S., Aitamurto, T.: Knapsack Voting for Participatory Budgeting. ACM Transactions on Economics and Computation **7**(2), 1–27 (2019)
9. Jain, P., Sornat, K., Talmon, N.: Participatory Budgeting with Project Interactions. In: Proceedings of the 29th International Joint Conference on Artificial Intelligence. pp. 386–392. ijcai.org (2020)
10. Kellerer, H., Pferschy, U., Pisinger, D.: Knapsack Problems. Springer (2004)
11. Krentel, M.W.: The Complexity of Optimization Problems. Journal of Computer and System Sciences **36**(3), 490–509 (1988)
12. Lackner, M., Skowron, P.: Approval-Based Multi-Winner Rules and Strategic Voting. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence. pp. 340–346. ijcai.org (2018)
13. Rey, S., Endriss, U., de Haan, R.: Designing Participatory Budgeting Mechanisms Grounded in Judgment Aggregation. In: Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning. pp. 692–702. ijcai.org (2020)
14. Rey, S., Endriss, U., de Haan, R.: Shortlisting Rules and Incentives in an End-to-End Model for Participatory Budgeting. In: Proceedings of the 30th International Joint Conference on Artificial Intelligence. ijcai.org (August 2021), to appear.
15. Sintomer, Y., Herzberg, C., Röcke, A.: Participatory Budgeting in Europe: Potentials and Challenges. International Journal of Urban and Regional Research **32**(1), 164–178 (2008)
16. Skowron, P., Faliszewski, P.: Chamberlin–Courant Rule with Approval Ballots: Approximating the MaxCover Problem with Bounded Frequencies in FPT Time. Journal of Artificial Intelligence Research **60**, 687–716 (2017)
17. Talmon, N., Faliszewski, P.: A Framework for Approval-based Budgeting Methods. In: Proceedings of the 33rd AAAI Conference on Artificial Intelligence. vol. 33, pp. 2181–2188. AAAI Press (2019)

CHAPTER 6

TIME-CONSTRAINED PARTICIPATORY BUDGETING UNDER UNCERTAIN PROJECT COSTS

In this chapter, we study a more realistic model for participatory budgeting by considering uncertain project costs and implementation durations for each project. In particular, we initiate an axiomatic study for our framework, extend existing notions of proportionality, and experimentally evaluate novel rules, designed to trade-off desirable properties.

6.1 Summary

In this work, we introduce a more general framework for participatory budgeting, to model some real-world applications more realistically. We assume that the cost for implementing a project is not fixed in advance, but rather given as an estimate. More precisely, as formalized in Subsection 2.3.2, the exact cost for each project is only determined after implementation, where the estimated cost is coming from a probability distribution with an upper and lower bound on the maximum and minimum cost. In real-world campaigns, projects often have to be implemented in a given time frame (e.g., a legislation period). Therefore, each project is additionally equipped with a duration, after which the exact cost is revealed (in case of implementation). Hence, an algorithm to decide on the implementation of projects can act in an online fashion, since after the implementation of a project more information about the exact cost (and the expected leftover budget) is derived.

We study our framework from an axiomatic and algorithmic point of view. First, we identify desirable properties, any budgeting procedure with uncertain project costs should satisfy. That is, we assume a reasonable online budgeting method should satisfy *punctuality* (not exceeding the time frame), *risk-assessment* (minimize the risk of exceeding the budget limit), *limitation* (limit the cost that might overshoot the budget), and *exhaustiveness* (budget should not remain unused if more projects can certainly be funded). As any such algorithm should still perform well in terms of social welfare, we additionally study the *competitive ratio* [82] of any online mechanism. That is, the worst-case ratio for the social welfare compared with an optimal offline algorithm, where the exact cost for each project is known in advance.

Our axiomatic analysis reveals the limits of compatibility for our axiomatic properties. In summary: Each pair of punctuality, risk-assessment, and exhaustiveness can be satisfied simultaneously by an online budgeting method, while it is generally impossible to satisfy all three. Subsequently, we introduce *best effort budgeting methods* as a way to trade-off incompatible axiomatic properties and evaluate these methods experimentally.

We complement our results by extending related research on proportionality [6, 142] for participatory budgeting to our setting with uncertain cost. In particular, we generalize related axioms to an *ex ante* and *ex post* variant and adapt the popular *Method of Equal Shares* [142] to derive a possibility result. That is, assuming maximum cost when necessary and exact cost when possible, our variation of the Method of Equal Shares satisfies risk-assessment, limitation, and *extended justified representation* [142] in an ex post (and thus, also ex ante) fashion.

6.2 Reflection on Initial Research Goals

In this work, we addressed three of our initial research questions, introduced in Chapter 3. We initiated the formal study on a more general framework for participatory budgeting with uncertain project costs and project durations. Most significantly, we addressed Question Q1, by *(i)* designing novel desirable axiomatic properties for our framework, *(ii)* generalizing well-known proportionality axioms to the uncertain context in an ex ante and ex post variant, and *(iii)* providing an extensive study on compatible and incompatible axioms. Further, we addressed Question Q2, by presenting polynomial-time computable best effort algorithms,[51] which always satisfy a compatible set of axioms, while trying their best to also satisfy an additional, but generally incompatible, axiom. Finally, we contributed to answering Question Q4, by generalizing the Method of Equal Shares to our framework, in order to satisfy ex post extended justified representation.

[51] Depending on the probability distribution, an output can only be approximated in polynomial time by using a sampling approach.

6.3 Publication

This work has been published and presented as a full paper at the *31st International Joint Conference on Artificial Intelligence*.

[24] D. Baumeister, L. Boes, and C. Laußmann. "Time-Constrained Participatory Budgeting Under Uncertain Project Costs". In: *Proceedings of the 31st International Joint Conference on Artificial Intelligence*. ijcai.org, 2022, pp. 74–80

Parts of this article, containing additional results regarding uncertainty on the projects' durations and more evolved insights into our experimental study, were published in Christian Laußmann's dissertation [121].

6.4 Personal Contribution

The initial idea of this work was inspired by an informal lunch conversation between Christian Laußmann and Jérôme Lang. Formally pursuing this idea, the conception and writing was conducted jointly with my co-authors Dorothea Baumeister and Christian Laußmann. Although the overall model and the axiomatic properties in Section 3 were designed by all contributing authors in equal parts, the technical results in this section were gathered in an insightful collaboration with Christian Laußmann by an equal share of work. The formulation of best effort budgeting methods and their experimental evaluation in Section 4 was mainly contributed by Christian Laußmann. In turn, the technical results on proportionality in Section 5, including the generalizations of both respective axioms and the Method of Equal Shares, were mostly contributed by me.

Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22)

Time-Constrained Participatory Budgeting Under Uncertain Project Costs

Dorothea Baumeister, **Linus Boes** and **Christian Laußmann**
Heinrich-Heine University Düsseldorf
{d.baumeister, linus.boes, christian.laussmann}@uni-duesseldorf.de

Abstract

In participatory budgeting the stakeholders collectively decide which projects from a set of proposed projects should be implemented. This decision underlies both time and monetary constraints. In reality it is often impossible to figure out the exact cost of each project in advance, it is only known after a project is finished. To reduce risk, one can implement projects one after the other to be able to react to higher costs of a previous project. However, this will increase execution time drastically. We generalize existing frameworks to capture this setting, study desirable properties of algorithms for this problem, and show that some desirable properties are incompatible. Then we present and analyze algorithms that trade-off desirable properties.

1 Introduction

As an introduction, we consider the following example, which will accompany us through our work.

Example 1. *The country "Participation Island" wants to address climate change. The government decides to provide* 200 *million dollars for new mobility projects that reduce emissions. Climate change will not wait forever, so these projects should be finished within the next five years. Companies send proposals for projects that each cost at most* 200 *million and can be realized within the given five years: bike sharing (BS); an express train route (ET); electric vehicle charging stations (EV); and development of fuel-cell vehicles (FV). The citizens of Participation Island are asked to vote on these projects by approving each project they like. The government then chooses which projects should be realized according to the voters' preferences and the constraints on time and money.*

This problem has been extensively studied in Knapsack and participatory budgeting literature. Yet, it is not realistic.

Example 1 (continued)**.** *While it is very predictable how much EV and BS will cost (because one knows the cost for each station/bike and the number of stations/bikes), there is some uncertainty about the other projects. It is known that the train route goes through undeveloped swamps, where it is unclear yet, how many foundations are needed. The company which proposed the project guarantees that the costs are between* 80 *and* 150 *million. Hydrogen technologies are still in the development and while the company is sure that they can develop the cars, they are unsure about the exact costs. They guarantee the costs to be between* 100 *and* 160 *million.*

The exact costs of a project are usually only known after finishing it. A common way to accommodate this uncertainty is to submit the project application with an *estimated* cost. However, if the estimation is far too expensive, the company will probably not get the bid. On the other hand, if it is far too low, the company is unable to realize the project and will have to ask for additional money.[1] So a more realistic approach is to provide a cost range, which moves the uncertainty risk to Participation Island. The question is now: *how should Participation Island select projects?* Of course, the limit of 200 million should not be exceeded, or at least this should be improbable. To reduce this risk, they could implement the projects sequentially, and wait for the exact costs before deciding which project to start next. But this is a slow process, and the time limit of five years should also not be exceeded.

Our Contribution. We develop a framework for time-constrained participatory budgeting under uncertain project costs based on the existing framework for approval-based participatory budgeting by Talmon and Faliszewski [2019]. We propose desirable properties for such budgeting methods and explore which properties can be combined and which are incompatible. In addition, we experimentally evaluate algorithms for uncertain project costs that sequentially start new projects using the information of the cost of already finished projects and thus optimize the utilization of the budget. Meanwhile, they do their *best effort* to satisfy as many desirable properties as possible. Furthermore, we analyze different forms of proportionality for uncertain project costs.

Related Work. First, our work is related to scheduling and project planning literature. Given a set of tasks (which may form a bigger project), this research area is about making a plan on how to complete *all* tasks while respecting their

[1] Depending on the contract, the company will have to pay from own resources. However, it is common that cost estimations may be exceeded by $10-20\%$ without further consultation with the client. And if the company has not enough own resources to pay for the extra cost, they may become bankrupt, and the client has to pay someone else for finishing the project.

dependencies and resource requirements. In the *Resource-Constrained Project Scheduling Problem* the goal is to minimize the completion time of the last project while projects have dependencies and resource requirements (e.g. workers), so parallelization of projects is not always possible. A similar problem is the *Time-Constrained Project Scheduling Problem* where the goal is to minimize the amount of extra resources needed to finish all projects in time. Both problems have been studied with uncertain project durations (see [Ma *et al.*, 2016] and [Moradi and Shadrokh, 2019]). However, the resource requirements are deterministic. Vaziri *et al.* [2007] analyze project planning where the time for tasks is uncertain and can be influenced by the resources allocated to that task.

Note that in comparison to the above problems, we do not aim at implementing all projects and we also want to take into account the voters' preferences. In fact, if there is enough budget available to implement all projects our problem is trivial. Instead, we want to implement a subset of projects that makes the stakeholders happy, while facing the budget and the time as resource constraints. This is related to project portfolio management where the question is *which* projects should be implemented, suspended, or canceled, to serve the overarching objective of an institution. For an introduction to project portfolio management see [Rad and Levin, 2006, Chapter 1]. Another difference is that projects in our model have no dependencies on other projects, and our resources money and time are bounded, unlike resources in scheduling.

Also related is the work of Pindyck [1993], who studies projects that require continuous investment, and the final cost is only known after completion. However, there is no time constraint, no preferences, and projects can be canceled.

Another closely related field is Knapsack, as we study maximizing the (additive) utility of a set of projects while facing a cost constraint. For a broad overview, we refer to the textbook by Kellerer *et al.* [2004], which includes a chapter on multidimensional Knapsack problems considering more than one resource (in our case cost and time). Setting aside the time dimension, Knapsack has been studied under uncertain weights similar to our model by Monaci and Pferschy [2013] and Monaci *et al.* [2013]. Following a concept by Bertsimas and Sim [2004], both aim to find *robust* solutions, that perform well even if the exact weights turn out to be unfavorable. The model by Goerigk *et al.* [2015] allows for querying the exact weight of a fixed number of items in order to find a good solution when weights are uncertain.

A collective variant of Knapsack, namely participatory budgeting, has gained some attention in computational social choice lately. For notation, we adopt the formal participatory budgeting framework for approval-based preferences, which was introduced by Talmon and Faliszewski [2019] and extended to irresolute budgeting rules by Baumeister *et al.* [2020]. We refer to the bookchapter by Aziz and Shah [2021] for a broad overview on participatory budgeting in the context of computational social choice. Gomez *et al.* [2016] present a broad model considering uncertainty for both, cost and utility, for every project. In contrast to ours, their model is purely stochastic (a set of projects is feasible if its expected cost is within the budget limit), and projects are implemented all at the same time. An important stream of research we follow is the concept of *proportionality* in collective decision making, where every voter should be represented equally by a given solution. Aziz *et al.* [2018] study a variety of axioms suitable for participatory budgeting, while Pierczyński *et al.* [2021] also provide a rule with desirable properties with respect to proportional representation. We generalize some of their results to work with uncertain costs.

2 Preliminaries

Throughout this paper, for $i, j \in \mathbb{N}$ we write $[i, j] = \{i, \dots, j\}$ and $[i] = [1, i]$. Let $A = \{a_1, \dots, a_m\}$ be the set of *projects* and each subset $B \subseteq A$ is a *bundle*.[2] In our model we assume uncertainty about the exact *cost* for every project. Therefore, each project is associated with a total of four cost functions $\widetilde{c} = (c_{\min}, c_{\max}, c, c_p)$, where $c_{\min}$ and $c_{\max}$ model lower and upper bounds on the project's costs, while c models the exact costs. Hence, all three functions map from A to $\mathbb{N}^+$ and for each project $a \in A$ it holds that $c(a) \in [c_{\min}(a), c_{\max}(a)]$. For simplicity, we abuse notation by denoting $c(B) = \sum_{a \in B} c(a)$ as the cost of a bundle B (analogously for $c_{\min}$ and $c_{\max}$). By $c_p(a, x)$ we denote the probability that project $a \in A$ costs at most $x \in \mathbb{N}^+$. Note that c_p is monotonic, $c_p(a, c_{\min}(a) - 1) = 0$, and $c_p(a, c_{\max}(a)) = 1$. Slightly abusing notation we write $c_p(B, x)$ to denote the probability that for a given bundle B the cost $c(B)$ is bounded by x. Finally, each project $a \in A$ takes time $\delta : A \to \mathbb{N}^+$ to finish, and we have an overall *time limit* $\tau \in \mathbb{N}^+$ at which all projects have to be finished, and a *budget limit* ℓ (also referred to as budget)[2], which is the available money to implement projects. We assume for no project $a \in A$ holds $\delta(a) > \tau$ or $c_{\max}(a) > \ell$.

The projects are evaluated by a set of *voters* $V = \{v_1, \dots, v_n\}$. Each voter v approves a subset of projects denoted by $\mathrm{app}_v \subseteq A$. By $s_v(B) = |B \cap \mathrm{app}_v|$ we denote the *satisfaction* of voter v with bundle B, i.e. the number of items in B approved by v.[3] We define $s(B) = \sum_{v \in V} s_v(B)$ as the *total satisfaction* of all voters. We assume $s(\{a\}) > 0$ for all $a \in A$, i.e., each project is approved by at least one voter.

Let $E = (A, V, \widetilde{c}, \ell, \delta, \tau) \in \mathcal{E}$ be a budgeting scenario with uncertain cost, where $\mathcal{E}$ is the set of all such scenarios. An *online budgeting method* $\mathcal{R}$ works in discrete time steps, and successively builds a *budgeting log* $L : A \to \mathbb{N} \cup \{\bot\}$ representing at which time step a project has been started, where $\bot$ denotes that the project will not be realized. The budgeting method has limited access to the cost function c. If $L(a) = t^*$, then $c(a)$ is available only after the project has been implemented, i.e. at time step $t^* + \delta(a)$. Obviously, the decision made at step t^* is fixed and may not be revised when more information is available. Formally, the output of a budgeting method $\mathcal{R}(E)$ is a budgeting log. Further, the set of *realized* projects for a budgeting log L is denoted by $R(L) = \{a \in A \mid L(a) \neq \bot\}$. On the other hand, an *offline* budgeting method may access the exact cost function

[2] We differ in terminology from Talmon and Faliszewski [2019], who refer to a project selection as *budget*.

[3] Also studied by Talmon and Faliszewski [2019]. Other satisfaction functions (e.g. cost based) do not fit varying costs that well.

Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22)

c. Hence, an optimal solution can be precomputed and every project can be implemented simultaneously.

Example 1 (continued). *We have $A = \{BS, ET, EV, FV\}$ with following costs (in million), durations, and satisfactions.*

a	$c(a)$	$c_{\min}(a)$	$c_{\max}(a)$	$\delta(a)$	$s(\{a\})$
BS	40	37	42	1	5,000
ET	120	80	150	4	9,000
EV	59	59	61	1	6,000
FV	100	100	160	5	11,000

The budget limit is $\ell = 200$ and we have time $\tau = 5$. If we knew the exact costs, we would immediately start the projects BS, EV, and FV since with cost of 199 they fit in our budget and have the maximum number of 22,000 approvals. However, this bares a high risk if the cost is unknown. The projects could also cost up to 263 which is far beyond our budget. An online budgeting method could for instance do the following. Start ET in the first year and wait four years until it finishes. If we are lucky, it turns out that it costs at most 97 so that we can safely implement both BS and EV in the remaining year. If it costs more, we can implement at least BS or in some cases EV. Also, the following is possible. If it is very improbable that FV costs more than 139, we could also relatively safely begin FV and EV simultaneously in the first year. Note that a sequential implementation of FV and EV fails the time limit.

We assume each started project in a budgeting log will be implemented regardless of the final cost and completion time. It is often assumed that projects can be stopped if their cost becomes more than their value, i.e. one cannot gain money with the project (e.g. in [Pindyck, 1993]). We decided against canceling for two reasons. First, in our model the "value" of a project is not measured in the same currency as the cost; second, the projects we have in mind cannot simply be canceled (imagine an eternal construction site in the city center).

3 Properties of Online Budgeting Methods

A budgeting log, and thus also an online budgeting method, has rather weak requirements. For instance, it is allowed in a budgeting log to start arbitrarily many projects simultaneously, even if they will certainly exceed the budget limit; or to start projects so late that they cannot be completed in time. These issues are undesirable and should be avoided. In this section, we define some desirable properties a budgeting log (and the algorithm that generates it) should satisfy.

Definition 1. *Let $E \in \mathcal{E}$ be a budgeting scenario and L be a budgeting log with respect to E. L satisfies the following axioms if respective conditions are met.*

Punctuality (PU): *Every realized project finishes within the given time limit. Formally, for all $a \in A$ it holds that either $L(a) = \bot$ or $L(a) + \delta(a) \leq \tau$.*

α-Risk-assessment (α-RA): *A (set of) project(s) may only be started if the probability for exceeding the budget limit is at most α. Formally, for every $t \in [\tau]$, let $U_t = \{a \in A \mid L(a) \leq t < L(a) + \delta(a)\}$ be the running yet unfinished projects, and $F_t = \{a \in A \mid L(a) + \delta(a) \leq t\}$ the finished projects. For given $\alpha \in [0, 1)$, it holds that a set of projects S may only be started at time t if $c_p(U_t \cup S, \ell - c(F_t)) \geq 1 - \alpha$.*

κ-Limitation (κ-LI): *The budget limit may not be exceeded by a factor greater than κ. Formally, $c(R(L)) \leq \kappa\ell$.*

Exhaustiveness (EX): *There should be no project, which could have been implemented even with maximum cost without breaking feasibility. Formally, for $B = R(L)$ and every $a \in A \setminus B$, it holds that $c(B) + c_{\max}(a) > \ell$.*

Note that 0-risk-assessment and 1-limitation coincide. A budgeting method $\mathcal{R}$ satisfies some axiom χ if $\mathcal{R}(E)$ satisfies χ for every $E \in \mathcal{E}$ (assuming parallel universe tie-breaking).

We study punctuality as a property since it is very interesting to see what kind of restriction it is, and what algorithms are possible if we relax it. Risk-assessment and limitation can be interpreted as follows. The client (e.g. Participation Island) has $(\kappa - 1)\ell$ extra money as a security — for instance as a loan option — which should be used only if absolutely necessary. With a good risk-assessment (i.e. small α) it is improbable that the security is ever touched. Exhaustiveness has two interpretations. First, voters naturally expect that approved projects are realized if there is money left to do so safely. Second, it is common that the budget of a department may be reduced in the next period if it is not completely spent.

Independent of the above properties we want to maximize the satisfaction of the voters with the outcome. One key metric for the analysis of online optimization algorithms is the worst-case ratio between a solution found by an online algorithm and an optimal (satisfaction maximizing) solution with complete knowledge. This factor is known as *competitive ratio* (CR) (see Fiat and Woeginger [1998]).

Definition 2. *An online budgeting method $\mathcal{R}$ is σ-competitive (σ-CR) if there is a constant $\Delta \in \mathbb{R}$, such that for every $E \in \mathcal{E}$ and $\mathcal{B}_\ell = \{B \subseteq A \mid c(B) \leq \ell\}$ it holds that $s(R(\mathcal{R}(E))) + \Delta \geq \frac{1}{\sigma} \max_{B \in \mathcal{B}_\ell} s(B)$.*

So which combinations of properties are possible, and is there a perfect online budgeting method? Unfortunately, the answer is no, i.e. no method can satisfy all properties simultaneously for all combinations of parameters.

Theorem 3. *For any fixed $\alpha < 1$, no online budgeting method simultaneously satisfies α-risk-assessment, punctuality, and exhaustiveness.*

Proof. Consider an odd budget limit $\ell \geq 3$, and $A = \{a_1, a_2, a_3, \ldots\}$. Each project $a_i \in A$ has minimum cost $c_{\min}(a_i) = {}^{(\ell-1)}/_2$, maximum cost $c_{\max}(a_i) = {}^{(\ell+1)}/_2$, and takes time $\delta(a_i) = \tau$. Note that a set of two projects exceeds the budget limit if and only if both projects have maximum cost. Let each project have maximum cost with probability greater than $\sqrt{\alpha}$. Due to α-risk-assessment a budgeting method can only start one project at the first time step, say a_1. By punctuality it is impossible to start another project. Since $c(a_1) = {}^{\ell}/_2$ is possible, there are instances for which $c(a_1) + c_{\max}(a_2) \leq \ell$, thus exhaustiveness is violated. □

Similar holds for κ-limitation as long as $\kappa < m = |A|$.

Theorem 4. *For any fixed $\kappa < m$, there exists no online budgeting method simultaneously satisfying κ-limitation, punctuality, and exhaustiveness.*

Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22)

Proof. Consider $E \in \mathcal{E}$ with $A = \{a_1, \ldots, a_m\}$, $m \geq 3$, and $\ell > m$. Each project $a_i \in A$ takes time $\delta(a_i) = \tau$ to realize, has cost $c(a_i) = 1$, and maximum cost $c_{\max}(a_i) = \ell - m$. By exhaustiveness, all projects must be realized, since for each $a_i \in A$ holds $c(A \setminus \{a_i\}) + c_{\max}(a_i) = \ell$ and by punctuality, all projects must be started simultaneously. However, this decision has to be made without knowing the exact cost. Let $E' \in \mathcal{E}$ be equivalent to E, except for having maximum cost as exact cost for each project. An online budgeting method that implements all projects to satisfy exhaustiveness and punctuality might end up spending $c_{\max}(A) = m \cdot (\ell - m)$. Thus, to start all projects, it cannot be better than $\frac{m \cdot (\ell - m)}{\ell} = \left(m - \frac{m^2}{\ell}\right)$-limited. By choosing ℓ large, we can approach m to any fixed value $\kappa < m$. □

Interestingly, for $\kappa \geq m$, above properties are compatible and can be satisfied by a 1-competitive algorithm. Since $c_{\max}(a) \leq \ell$ holds for every $a \in A$, we can implement every project at the first time step, only violating risk-assessment.

Observation 5. *There exists a* 1*-competitive online budgeting method satisfying* m*-limitation, punctuality, and exhaustiveness.*

However, trading off some desirable properties, it is possible to achieve 0-risk-assessment (or equivalently 1-limitation) together with either punctuality or exhaustiveness.

Theorem 6. *There is an* m*-competitive online budgeting method satisfying* 0*-risk-assessment (and thus* 1*-limitation) and either punctuality or exhaustiveness.*

Proof. First, we start the most valuable project, say a_1, which has by definition maximum cost of at most ℓ. This way we achieve m-competitiveness already, since $s(A) \leq m \cdot s(\{a_1\})$. If we want to achieve punctuality, we stop now. For exhaustiveness, we sequentially add those projects, that can be safely added without exceeding the budget. □

A competitive ratio of m is bad. Yet, for fixed α this factor cannot be improved if the projects' cost intervals are large.

Theorem 7. *For any online budgeting method that satisfies* α*-risk-assessment for a fixed* $\alpha < 1$*, the competitive ratio is in* $\Omega(m)$*. If* $c_{\max}(a) = c_{\min}(a) + 1$ *holds for all* $a \in A$*, the competitive ratio is in* $\Omega(2)$*.*

Proof. Consider $E \in \mathcal{E}$ with $A = \{a_1, a_2, a_3, \ldots, a_m\}$ and a set of voters, such that each project $a_i \in A$ yields the same (additive) satisfaction $s(a_i) = \lambda \in \mathbb{N}^+$. Let $\ell = m$, $c_{\min}(a_i) = 1$ and $c_{\max}(a_i) = m$ for every $a_i \in A$. Further, for each a_i we set the probability that a_i costs exactly m to α (thus, $c_p(a_i, \ell - 1) = 1 - \alpha$). An online algorithm with α-risk-assessment cannot start more than one project at the same time because for every pair of projects $a_i \neq a_j$ it holds $c_p(\{a_i, a_j\}, \ell) \leq c_p(a_i, \ell - 1) \cdot c_p(a_j, \ell - 1) = (1 - \alpha)^2 < 1 - \alpha$. Thus it starts at most one project, for example a_1. Revealing $c(a_1) = m$ and $c(a_i) = 1$ for $i \in [2, m]$, an offline algorithm may select the optimal solution $B = A \setminus \{a_1\}$ with $s(B) = \lambda \cdot (m - 1)$, while the online algorithm yields a satisfaction of $s(\{a_1\}) = \lambda$. Overall we deduce a competitive ratio of $(m - 1) \in \Omega(m)$.

Properties	CR	Ref
PU, EX, m-LI	1-CR	Obs. 5
0-RA, 1-LI, PU	m-CR (up to 2-CR)	Thm. 6, 8
0-RA, 1-LI, EX	m-CR (up to 2-CR)	Thm. 6, 8

Table 1: Summary of our possibility results regarding the combination of axioms for online budgeting methods.

Property	Incompatible	Ref
PU	$\{\alpha$-RA, EX$\}$, $\{m'$-LI, EX$\}$	Thm. 3, 4
α-RA	{PU, EX}	Thm. 3
m'-LI	{PU, EX}	Thm. 4
EX	$\{\alpha$-RA, PU$\}$, $\{m'$-LI, PU$\}$	Thm. 3, 4

Table 2: Summary of our impossibility results regarding the combination of axioms for online budgeting methods ($m' < m$).

For bounded uncertainty by $c_{\max}(a) = c_{\min}(a) + 1$ for all $a \in A$, we can use a similar argument. Let $\ell \geq 2$ be an even number, $A = \{a_1, a_2, a_3, \ldots, a_m\}$ with $c_{\min}(a_i) = \ell/2$ and $c_{\max}(a_i) = \ell/2 + 1$. Again, we assume equal utility of λ for every project. We set $c_p(a_i, \ell/2) = 1 - \alpha$, so the probability that two projects can be implemented within ℓ is $(1 - \alpha)^2 < 1 - \alpha$. Let an online budgeting method implement a_1 first (due to α-RA it cannot implement two projects). Revealing $c(a_1) = \ell/2 + 1$ and $c(a_2) = c(a_3) = \ell/2$, the optimal solution is $\{a_2, a_3\}$, yielding a competitive ratio of $\frac{2\lambda}{\lambda}$. □

Theorems 6 and 8 show that these bounds are tight.

Theorem 8. *If the uncertainty on the cost is bounded by a small factor* c^**, that is,* $c_{\max}(a) - c_{\min}(a) < \frac{c_{\max}(a')}{m} = c^*$ *for all* $a, a' \in A$*, there is a* 2*-competitive method satisfying* 0*-risk-assessment and either punctuality or exhaustiveness.*

Proof. We use the optimal offline method to retrieve an optimal bundle B, assuming the lower bound cost for each project, i.e. $\sum_{b \in B} c_{\min}(b) \leq \ell$.
Case 1: If $\sum_{b \in B} c_{\max}(b) \leq \ell$, we are done.
Case 2: Otherwise, since implementing B may exceed the budget limit, we remove the least valuable project $a \in B$ and implement $B' = B \setminus \{a\}$ at the first time step. It holds that $\ell \geq \sum_{b \in B} c_{\min}(b) \geq \sum_{b \in B} c_{\max}(b) - |B| c^* \geq \sum_{b \in B} c_{\max}(b) - c_{\max}(a) = \sum_{b \in B'} c_{\max}(b)$, due to $|B| c^* \leq |B| \cdot \frac{c_{\max}(a)}{m} \leq c_{\max}(a)$. On the other hand, since a is the least valuable project in B, the satisfaction with B' is at least $s(B') \geq s(B) \cdot \frac{|B'|}{|B|} = s(B) \cdot \frac{|B|-1}{|B|}$. The worst competitive ratio of two is achieved if $|B| = 2$, since $|B| = 1$ is already covered by case 1. We can now return B' which satisfies punctuality or we add projects to B' until we achieve exhaustiveness. Note that in both cases there is no risk for exceeding the budget limit. □

Table 1 summarizes the lower bound competitive ratios any online algorithm can achieve while satisfying given axiomatic properties, while Table 2 summarizes incompatible axioms.

Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22)

4 Best Effort Online Budgeting

The possibility and impossibility results show that there is no perfect online budgeting method, i.e., no method satisfies all properties and has a good competitiveness. So the only way is to design "best effort" algorithms that trade-off properties and have a good competitiveness in most cases.

We propose the online budgeting method **Best effort exhaustiveness (BEE)** that trades exhaustiveness against punctuality, risk-assessment, and limitation. That is, the method guarantees punctuality, α-risk-assessment, and κ-limitation for given τ, α, κ, but not exhaustiveness. However, it tries to be as exhaustive as possible. With small modifications, we get the **Best effort punctuality (BEP)** method which trades punctuality against exhaustiveness, risk-assessment, and limitation. Both algorithms generalize a common greedy algorithm for Knapsack (see [Kellerer *et al.*, 2004]) to our setting.

We test both algorithms using real data from the Participatory Budgeting Library (see [Stolicki *et al.*, 2020]), modified to fit our uncertainty scenario. BEE performs better in terms of both exhaustiveness and competitiveness the more we increase τ. If projects have durations in $[1, 10]$, the performance is already remarkably good at time limits between 20 and 30. However, our experiments with the BEP method imply that guaranteed exhaustiveness results in massive unpunctuality. Also, κ-limitation plays a role for the exhaustiveness, however, the role of α-risk-assessment is negligible unless α is almost 0.

5 Proportionality

Apart from maximizing the overall utility, another way to satisfy voters is a proportional distribution of the realized projects among them. There is a variety of proportionality axioms in the literature. We focus on *justified representation*, considering *BPJR-L* by Aziz *et al.* [2018] and *Extended Justified Representation* by Pierczyński *et al.* [2021]. Assuming uncertainty over the exact projects' costs, we provide two relaxations for proportionality axioms: *ex ante* and *ex post*. For the relaxations, we assume the upper cost bound is given for all projects (ex ante) or not implemented projects (ex post).

Definition 9. *Consider a budgeting scenario $E = (A, V, \widetilde{c}, \ell, \delta, \tau) \in \mathcal{E}$ and a bundle $B \subseteq A$. We define the following axioms in different variants, distinguishable by respective cost functions. Let $c' : A \to \mathbb{N}^+$ be a cost function. Typically, for known exact cost, we consider $c' = c$ for both of the following axioms. We study two relaxations for our setting with uncertain cost. That is, for the* ex ante *variant, we study $c' = c_{\max}$ and for the* ex post *variant, for $a \in A$ we study $c'(a) = c(a)$ if $a \in B$ and $c'(a) = c_{\max}(a)$, otherwise.*

A bundle B satisfies following axioms if the respective condition holds, while a budgeting method $\mathcal{R}$ satisfies an axiom if it holds for every $R(\mathcal{R}(E))$ with $E \in \mathcal{E}$.

BPJR-L: *For all $k \in [\ell]$ there exists no set of voters $V' \subseteq V$ with $\frac{|V'|}{n} \geq \frac{k}{\ell}$ such that $c'\left(\bigcap_{v \in V'} \mathrm{app}_v\right) \geq k$, but there is a set $T \subseteq \bigcap_{v \in V'} \mathrm{app}_v$ with $c'\left(\left(\bigcup_{v \in V'} \mathrm{app}_v\right) \cap B\right) < c'(T) \leq \frac{\ell \cdot |V'|}{n}$.*

Extended Justified Representation (EJR): *For all $V' \subseteq V$ and $T \subseteq \bigcap_{v \in V'} \mathrm{app}_v$ it holds that, if $c'(T) \leq \frac{\ell \cdot |V'|}{n}$, then there exists some $v \in V'$ with $|\mathrm{app}_v \cap B| \geq |T|$.*

Informally, a group of voters should not be worse off, if they could spend their proportional share of the budget on projects they collectively approve of. If every voter had exactly ℓ/n to spend, BPJR-L states that if a group of voters can afford a set of unanimously approved projects T, the funds spent on projects no one of the group approves should be lower than $\ell - c'(T)$. EJR on the other hand states that not all voters of a group should get less projects implemented than they could afford with joint funds.

Other axioms can be defined analogously in an ex ante and ex post version. In many cases stronger variants imply weaker variants. In case of EJR, consider some bundle that satisfies EJR. If the cost function is altered in a way that items only become more expensive, we observe that EJR is still satisfied (even though the price increase may now exceed the budget). Surprisingly, similar implications do not hold for BPJR-L.

Observation 10. *EJR implies ex post EJR and ex post EJR implies ex ante EJR.*

Theorem 11. *BPJR-L does not imply ex post (or ex ante) BPJR-L.*

Proof. Consider $E \in \mathcal{E}$ with three projects $A = \{a_1, a_2, a_3\}$, two voters $V = \{v_1, v_2\}$, and $\ell = 3$. The first two projects are known to be unit cost in advance, i.e., $c(a_i) = c_{\max}(a_i) = 1$ for $i \in [2]$. For a_3 it holds that $c(a_3) = 2$ and $c_{\max}(a_3) = 3$. For $i \in [2]$, voter v_i approves both a_i and a_3.

First, we will show that the bundle $B = \{a_1, a_2\}$ satisfies BPJR-L. There are three nonempty subsets of voters $V_1 = \{v_1\}$, $V_2 = \{v_2\}$ and $V_3 = \{v_1, v_2\}$. Let $T_i = \bigcap_{v \in V_i} \mathrm{app}_v$ and $U_i = \bigcup_{v \in V_i} \mathrm{app}_v$. For $i \in [2]$ it holds that $c(T_i) = c(U_i \cap B) = 1$. For V_3 it holds that $c(T_3) = c(U_3 \cap B) = 2$. Yet, ex post (and ex ante) BPJR-L is violated. For $k = 3$ it holds that $|V_3| = \frac{kn}{\ell}$, $c_{\max}(T_3) = 3 = k$ but $c(U_3 \cap B) = c_{\max}(U_3 \cap B) = 2 < k$. $\square$

Theorem 12. *There is no online budgeting method satisfying α-risk-assessment and ex post BPJR-L.*

Proof. Consider two projects $A = \{a_1, a_2\}$ with $c_{\min}(a_i) = 1$ and $c_{\max}(a_i) = 2$ for $i \in [2]$. Further, let $c_p(a_i, 1) < 1 - \alpha$ and $\ell = 2$. If some budgeting method $\mathcal{R}$ selects both projects simultaneously, α-risk-assessment is violated. If $\mathcal{R}$ selects one project $a \in A$ first, revealing $c(a) = 1$ yields two options. $B = \{a\}$ violates BPJR-L and implementing the remaining project violates α-risk-assessment. $\square$

To compute a feasible outcome satisfying an ex ante proportionality axiom, we can assume maximum cost for each project and compute a bundle that satisfies the strong variant of the axiom. For the actual cost the result may not be exhaustive, but we may implement it instantly. Aziz *et al.* [2018] showed that a feasible outcome satisfying BPJR-L is guaranteed to exist (although hard to compute). Regarding EJR, Peters and Skowron [2020] recently introduced an aggregation method for committee elections, called *Rule X* and Pierczyński *et al.* [2021] showed that a generalization of Rule X for participatory budgeting satisfies EJR (although they consider cardinal utilities instead of approval based preferences).

Observation 13. *There is an online budgeting method, satisfying 0-risk-assessment, punctuality and ex ante BPJR-L (respectively ex ante EJR).*

We slightly adjust Rule X to also work with uncertain cost and show subsequently that our variant satisfies ex post EJR.

Definition 14. *Rule X for uncertain cost ($\mathcal{R}_X$) works as follows. Every voter $v \in V$ is given a (real valued) individual budget of $b_v = \ell/n$, which they can use to implement projects sequentially. We start with an empty bundle $B = \emptyset$ and fund exactly one project in each iteration. The cost will be deducted from supporting voters' funds. For $\rho_{\max} > 0$ a project $a \in A \setminus B$ is $\rho_{\max}$-affordable if the following equation holds.*

$$\textstyle\sum_{v \in V | a \in \mathrm{app}_v} \min(b_v, \rho_{\max}) = c_{\max}(a) \qquad (1)$$

$\mathcal{R}_X$ implements the project a^* with the lowest $\rho_{\max}$-affordability, waits for it to finish to obtain the exact cost, and finally withdraws the required funds from approving voters. That is, we replace the upper cost bound $c_{\max}(a^*)$ in Equation (1) with $c(a^*)$ and calculate the ρ-affordability for a^*. Then for every voter $v \in V$ with $a^* \in \mathrm{app}_v$, b_v is set to $\max(0, b_v - \rho)$. If there is no $\rho_{\max}$-affordable project left, Rule X returns the corresponding budgeting log.

By definition, $\mathcal{R}_X$ always satisfies 0-risk-assessment (and 1-limitation) but fails punctuality. Following Pierczyński *et al.* [2021], Rule X (and thus $\mathcal{R}_X$) fails exhaustiveness.

Theorem 15. *$\mathcal{R}_X$ satisfies ex post EJR (and ex ante EJR).*

Proof. Let $E = (A, V, \widetilde{c}, \ell, \delta, \tau) \in \mathcal{E}$ and assume that $B = R(\mathcal{R}_X(E))$ violates ex post EJR. Consider the cost function c' with $c'(a) = c(a)$ if $a \in B$ and $c'(a) = c_{\max}(a)$ if $a \in A \setminus B$. Then by assumption there is a set of voters $V' \subseteq V$ and a set of projects $T \subseteq \bigcap_{v \in V'} \mathrm{app}_v$ with $c'(T) \leq \frac{\ell \cdot |V'|}{n}$ and for all $v \in V'$ it holds that $|\mathrm{app}_v \cap B| < |T|$. We investigate what led to the violation of EJR, by simulating $\mathcal{R}_X$ with (ex post) knowledge of the exact cost for projects in B. We index the elements in $T \setminus B = \{t_1, \dots, t_k\}$ in a way that $c'(t_i) \leq c'(t_j)$ for all $i < j$. Let every voter $v \in V'$ split her initial individual budget $b_v = \ell/n$ into $k+1$ piles b_v^j, such that for $j \in [k]$ the pile $b_v^j = \frac{c'(t_j)}{|V'|}$ is v's equal share for funding t_j (w.r.t. V'). The leftover funds $b_v^0 = \frac{\ell}{n} - \frac{c'(T)}{|V'|}$ will be reserved to fund projects in $B \cap T$. We execute $\mathcal{R}_X$ again, but this time we try to keep track of which project is financed with which pile of funds. We will show, that after execution either some voter $v \in V'$ has no funds left, but at least $|T|$ preferred projects in the outcome, or every voter has some pile left, which could have been used to implement a project in $T \setminus B$.

By assumption, each voter $v \in V'$ helps funding $T \cap B$ and at most $k-1$ additional projects. We let each voter $v \in V'$ pay for projects in $T \cap B$ with the leftover funds b_v^0. In the following numbering we skip projects in T (each voter in V' has reserved funds for $T \cap B$ and $T \setminus B$ wont be budgeted). For each $v \in V'$, let a_v^j be the j-th project, v helps funding (in addition to $T \cap B$). We let v pay her share for a_v^j with b_v^j. Note that $c_{\max}(a_v^j) \leq b_v^j$. Otherwise, either t_j has a lower $\rho_{\max}$-affordability and would have been funded instead, or some voter $v' \in V'$ has not enough budget left to pay her (full) share. In the latter case, consider the first iteration that leads to some $v' \in V'$ being bankrupt. Then v' has paid all the other projects with respective dedicated piles, resulting in at least k projects being funded (in addition to the projects $T \cap B$). This is a contradiction to EJR being violated, since $|\mathrm{app}_{v'} \cap B| \geq k + |T \cap B| = |T|$. Following $c_{\max}(a_v^j) \leq b_v^j$, our assumption can only hold if every voter $v \in V'$ has at least one pile untouched. This is a contradiction, as $\mathcal{R}_X$ could implement project t_1 with a total of $|V'|$ piles. □

Punctuality cannot be added without violating ex post EJR.

Theorem 16. *There is no online budgeting method satisfying α-risk-assessment, punctuality and ex post EJR.*

Proof. Consider the following example with $A = \{a_1, a_2\}$ with $c_{\min}(a_i) = 1$ and $c_{\max}(a_i) = 2$ for $i \in [2]$. The budget limit is set to $\ell = 3$ and a single voter approves both projects. We choose c_p, such that $c_p(A, 3) < 1 - \alpha$. Finally, projects need to be implemented at the first time step, i.e., $\delta(a_i) = \tau$ for $i \in [2]$. Let $B \subseteq A$ be the realized projects by an online budgeting method $\mathcal{R}$. We study three cases assuming punctuality is satisfied. If $|B| = 0$, B clearly fails ex post EJR. If $|B| = 1$, B might fail ex post EJR in case $c(B) = 1$. $\mathcal{R}$ may not select $B = A$ while respecting α-risk-assessment, as the probability of $c(A) = 4$ is greater than α. □

6 Conclusions And Outlook

In a participatory budgeting campaign with uncertain costs one has to trade-off between the desirable properties punctuality, exhaustiveness, risk-assessment, limitation, but also voter satisfaction and proportionality. In a way, this confirms the old project management principle "fast, cheap, good, pick two." Being punctual and within the cost limit makes proportionality or a good competitive ratio usually impossible.

In some applications, projects compete for resources like machines or workers. Thus not all projects can run simultaneously (see Hans *et al.* [2007]). As a next step, we propose to extend our model in that direction. Additionally, more realistic satisfaction functions could model that a voter's satisfaction with a project may be affected by discovering the exact cost. Dependencies between parameters could be especially interesting for time and money, allowing for the possibility to speed up a project by spending more money. Finally, our model only considers uncertainty in one of two resources. Swapping respective functions to study uncertain finishing times instead, we are interested in which implications still hold (although uncertainty concepts for proportionality are more reasonable for the cost dimension). A more general model may explore uncertainty of both dimensions.

Acknowledgements

The authors would like to thank Jérôme Lang and the anonymous reviewers for their helpful comments. This work was supported in part by the project "Online Participation," funded by the NRW Ministry for Innovation, Science, and Research and by the DFG grants RO-1202/21-1 and BA-6270/1-1.

References

[Aziz and Shah, 2021] Haris Aziz and Nisarg Shah. Participatory budgeting: Models and approaches. In T. Rudas and P. Gábor, editors, *Pathways between Social Science and Computational Social Science: Theories, Methods and Interpretations*, pages 215–236. Springer, 2021.

[Aziz *et al.*, 2018] Haris Aziz, Barton E. Lee, and Nimrod Talmon. Proportionally representative participatory budgeting: Axioms and algorithms. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, pages 23–31. IFAAMAS, 2018.

[Baumeister *et al.*, 2020] Dorothea Baumeister, Linus Boes, and Tessa Seeger. Irresolute approval-based budgeting. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*, pages 1774–1776. IFAAMAS, 2020.

[Bertsimas and Sim, 2004] Dimitris Bertsimas and Melvyn Sim. The price of robustness. *Operations research*, 52(1):35–53, 2004.

[Fiat and Woeginger, 1998] Amos Fiat and Gerhard J. Woeginger. *Online Algorithms: The State of the Art*, volume 1442. Springer, 1998.

[Goerigk *et al.*, 2015] Marc Goerigk, Manoj Gupta, Jonas Ide, Anita Schöbel, and Sandeep Sen. The robust knapsack problem with queries. *Computers & Operations Research*, 55:12–22, 2015.

[Gomez *et al.*, 2016] Javier Gomez, D. Rios Insua, and César Alfaro. A participatory budget model under uncertainty. *European Journal of Operational Research*, 249(1):351–358, 2016.

[Hans *et al.*, 2007] Erwin W. Hans, Willy Herroelen, Roel Leus, and Gerhard Wullink. A hierarchical approach to multi-project planning under uncertainty. *Omega*, 35(5):563–577, 2007.

[Kellerer *et al.*, 2004] Hans Kellerer, Ulrich Pferschy, and David Pisinger. *Knapsack Problems*. Springer, 2004.

[Ma *et al.*, 2016] Weimin Ma, Yangyang Che, Hu Huang, and Hua Ke. Resource-constrained project scheduling problem with uncertain durations and renewable resources. *International Journal of Machine Learning and Cybernetics*, 7(4):613–621, 2016.

[Monaci and Pferschy, 2013] Michele Monaci and Ulrich Pferschy. On the robust knapsack problem. *SIAM Journal on Optimization*, 23(4):1956–1982, 2013.

[Monaci *et al.*, 2013] Michele Monaci, Ulrich Pferschy, and Paolo Serafini. Exact solution of the robust knapsack problem. *Computers & Operations Research*, 40(11):2625–2631, 2013.

[Moradi and Shadrokh, 2019] Hadi Moradi and Shahram Shadrokh. A robust scheduling for the multi-mode project scheduling problem with a given deadline under uncertainty of activity duration. *International Journal of Production Research*, 57(10):3138–3167, 2019.

[Peters and Skowron, 2020] Dominik Peters and Piotr Skowron. Proportionality and the limits of welfarism. In *Proceedings of the 21st ACM Conference on Economics and Computation*, pages 793–794, 2020. Extended version arXiv:1911.11747.

[Pierczyński *et al.*, 2021] Grzegorz Pierczyński, Piotr Skowron, and Dominik Peters. Proportional participatory budgeting with additive utilities. *Advances in Neural Information Processing Systems*, 34, 2021.

[Pindyck, 1993] Robert S. Pindyck. Investments of uncertain cost. *Journal of Financial Economics*, 34(1):53–76, 1993.

[Rad and Levin, 2006] Parviz F. Rad and Ginger Levin. *Project Portfolio Management: Tools and Techniques*. International Institute for Learning, 2006.

[Stolicki *et al.*, 2020] Dariusz Stolicki, Stanisław Szufa, and Nimrod Talmon. Pabulib: A participatory budgeting library. *arXiv preprint arXiv:2012.06539*, 2020.

[Talmon and Faliszewski, 2019] Nimrod Talmon and Piotr Faliszewski. A framework for approval-based budgeting methods. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, volume 33, pages 2181–2188. AAAI Press, 2019.

[Vaziri *et al.*, 2007] Kabeh Vaziri, Paul G. Carr, and Linda K. Nozick. Project planning for construction under uncertainty with limited resources. *Journal of Construction Engineering and Management*, 133(4):268–276, 2007.

CHAPTER 7

COMPLEXITY OF SEQUENTIAL RULES IN JUDGMENT AGGREGATION

In this chapter, we explore a large class of (consistent) sequential judgment aggregation rules. In particular, we study the computational complexity for winner determination and manipulative interference, as well as their relationship to non-sequential rules.

7.1 Summary

In this work, we consider a large class of sequential judgment aggregation rules, as a generalization of the ranked agenda rule (see Lang and Slavkovik [118]). Using any resolute, complete, and complement-free (but possibly inconsistent) judgment aggregation rule $\mathcal{K}$, we construct the sequential rule $S\mathcal{K}$, which selects a consistent outcome as follows. Given a profile P of individual judgments and a predefined order $\pi = (\varphi_1, \ldots, \varphi_m)$ over the positive agenda Φ^+, $S\mathcal{K}$ decides sequentially in said order π whether an agenda item $\varphi_i \in \Phi^+$ or its complement should be added to the outcome. This is done by deciding whether one of those two items must be added to the outcome in order to remain consistency. If this is not the case, the underlying rule $\mathcal{K}$ decides on the acceptance of an issue, i.e., $\{\varphi_i, {\sim}\varphi_i\} \cap \mathcal{K}(P)$ is added to the outcome. Overall, we investigate questions related to winner determination and manipulative design from a computational complexity point of view. We complement our results by embedding our sequential rules profoundly into existing judgment aggregation literature, by establishing connections to other well-studied non-sequential rules.

In particular, for winner determination we show that determining whether a given formula is contained in the outcome $S\mathcal{K}(P, \pi)$ is in Δ_2^{P} for any efficiently computable judgment aggregation rule $\mathcal{K}$. For a large subclass of quota rules, introduced by Dietrich and List [66], we show that this bound is tight.

To study manipulative design, we explore an impossibly result by List [125], stating that the output of anonymous sequential rules (i.e., judges are treated equally) depends on the processing order over the issues. Hence, someone in control of this order might be able to alter the outcome of a sequential rule strategically (to better align with her preference). Furthermore, we study complementing problems regarding robustness. That is, asking whether a desired subset of issues appears in the outcome regardless of the underlying processing order. The upper bound complexity for both problems and a generic sequential rule range from a trivial polynomial-time computable algorithm to membership in the second level of the polynomial hierarchy (depending on whether the set of desired formulas consists of an individual judgment or a single formula). To show that those established bounds are tight for sequential quota rules, we formally design a novel counting technique (inspired by the famous technique used by Cook [57]) to model a Boolean formula that evaluates to TRUE if and only if the number of variables set to TRUE is bounded upwards by a given parameter.

Lastly, we show that the outcome of a sequential rule does not change, if all issues supported by the underlying rule are permuted to the beginning of the processing order. This allows us to connect our sequential rules to popular non-sequential rules, namely the max-card subagenda rule and the maximum subagenda rule [118] (and its generalization not based on majority support). Notably, our results on the computational complexity of manipulative design for sequential rules transfer directly to (irresolute) winner determination problems for the (generalized) maximum subagenda rule.

7.2 Reflection on Initial Research Goals

This article contributes to answering four of our initial research questions, introduced in Chapter 3. First, we addressed Question Q2 comprehensively, by *(i)* introducing sequential rules in a generic way, using any complete and complement-free rule as a blueprint, and *(ii)* providing general upper bounds and matching lower bounds for a large class of quota rules, using only two judges. Additionally, *(iii)* we generalized the maximum subagenda rule and transferred some of our complexity results to a corresponding winner determination problem. Second, we addressed Question Q3 by initiating a structured study on manipulative design, which arises from relying on a processing order to determine the outcome (as pointed out by List [125]). Overall, for a total of five related decision problems, we provided general upper bounds and matching lower bounds for quota rules, each falling into a different complexity class. We briefly touched Question Q1 by observing that the outcome of a sequential rule does not change, if the issues that are accepted by the underlying rule are permuted to the beginning of an order. Finally, this allowed us to deal with Question Q4 by showing that for an irresolute variant, our family of sequential rules coincides with the class of generalized maximum subagenda rules (inside the isolated field of judgment aggregation).

7.3 Publication

This work has been published and presented as a full paper at the *20th International Conference on Autonomous Agents and Multiagent Systems*.

[13] 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

The above publication is also contained in Robin Weishaupt's dissertation [174]. A related version of this article has been accepted for presentation at the *8th International Workshop on Computational Social Choice* after undergoing a review process.

[14] D. Baumeister, L. Boes, and R. Weishaupt. "Complexity of Sequential Rules in Judgment Aggregation". In: *The 8th International Workshop on Computational Social Choice*. Ed. by B. Zwicker and R. Meir. Haifa, Israel, 2021

7.4 Personal Contribution

Initiated by an open research question, proposed by Dorothea Baumeister, the conception and writing of this work was done jointly with my co-authors Dorothea Baumeister and Robin Weishaupt. All technical results were acquired in fruitful discussions with Robin Weishaupt by an equal share of work.

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

[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]).

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.

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 $\{1, \dots, r\}$. The judges judge over an **agenda** Φ, which consists of boolean formulas in standard propositional logic. In order to avoid double negations let $\sim\varphi$ denote the complement of φ, i.e., $\sim\varphi = \neg\varphi$ if φ is not negated, and $\sim\varphi = \psi$ if $\varphi = \neg\psi$. Thereby, we assume Φ 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 Φ 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 Φ. We split the agenda Φ into two disjoint subsets Φ_+ and Φ_-, 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 Φ. 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 Φ by $\mathcal{J}(\Phi)$. For the set of judges $[r]$ we denote their **profile** of individual judgments over Φ as $P = (P_1, \dots, P_r) \in \mathcal{J}(\Phi)^r$. We define a (resolute) **judgment aggregation rule** for an agenda Φ 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 Φ. 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., $R(P) = R(P_{\pi(1)}, \dots, P_{\pi(r)})$ 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) $\forall\varphi \in \Phi_+ : \varphi \in F_q(P) \Leftrightarrow |\{i \in [r] \mid \varphi \in P_i\}| \geq \lceil q(r + 1)\rceil$ and
(2) $\forall\varphi \in \Phi_- : \varphi \in F_q(P) \Leftrightarrow |\{i \in [r] \mid \varphi \in P_i\}| \geq \lfloor(1-q)(r+1)\rfloor$.

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 φ the quota rule executes some tie-breaking mechanism by choosing the corresponding formula from Φ_-, 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 Φ be an agenda, $P \in \mathcal{J}(\Phi)^r$ a profile and $\pi = (\varphi_1, \dots, \varphi_m)$ an order over Φ_+. 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 $(\varphi_1^* \wedge \dots \wedge \varphi_{i-1}^*) \models \varphi_i$ or $(\varphi_1^* \wedge \dots \wedge \varphi_{i-1}^*) \models \sim\varphi_i$ 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 φ_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 Φ 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 Φ 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 Φ_+. 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 Φ be an agenda, $P \in \mathcal{J}(\Phi)^r$ a profile for r judges, π an order over Φ_+ 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 π was processed until, but excluding ψ or $\sim\psi$ respectively.

Observe that for every $\psi \in \Phi$ either ψ itself or $\sim\psi$ appears in π, 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 $\leq_{\mathrm{m}}^{\mathrm{P}}$. SAT denotes the satisfiability problem and $\overline{\mathrm{SAT}}$ 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 Θ_2^p-complete for the distance-based procedure. Endriss and de Haan [8] showed that the winner problem is Θ_2^p-complete for some JA rules related to known voting rules (e.g. the maxcard rule), Δ_2^P-complete for the ranked agenda rule with a fixed tie-breaking and Σ_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 Θ_2^p (e.g. for the maxcard rule) and Π_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 $\mathcal{SK}$ is as follows.

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

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

Theorem 3.1. *$\mathcal{SK}$-Winner is in Δ_2^p if $\mathcal{K}$ is efficiently computable.*

Proof. Let $\mathcal{I} = (\Phi, P, \pi, \varphi)$ be a $\mathcal{SK}$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 φ_1 we will use the result of $\mathcal{K}$ based on P to decide whether to add φ_1 or $\sim\varphi_1$ to $\mathcal{SK}(P, \pi)$. Furthermore, denote by $\varphi_1^*, \ldots, \varphi_{i-1}^*$ the elements added to $\mathcal{SK}^{\varphi_i}(P, \pi)$ in the first $i-1$ iterations. Note, that we add any φ'_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 $\mathcal{SK}(P, \pi)$, since it is irrelevant for our purpose whether φ'_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 Δ_2^p. Thereby, it follows that $\mathcal{SK}$W$\in \Delta_2^p$ holds. ◻

In the construction above all queries rely on previous iterations and therefore, cannot be parallelized. Hence, Θ_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 Δ_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 α satisfiable and $x_n = 1$ in α'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 Δ_2^p-complete.*

Proof. From the previous theorem we know that $SF_q\mathrm{W} \in \Delta_2^p$ holds, since F_q is efficiently computable, complement-free and complete. Therefore, it is sufficient to show OMS $\leq_m^p SF_q$-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_qW 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 β_1, β_2, and γ are new variables, and $\alpha' = (\alpha \wedge \gamma) \vee \neg\beta_1 \vee \neg\beta_2$. Furthermore, we define the order π over Φ_+ as $\pi = (\beta_1, \beta_2, \alpha', \alpha' \wedge x_1, \ldots, \alpha' \wedge x_n)$ and the judges' profile P as follows.

P	β_1	β_2	α'	$\alpha' \wedge x_1$	$\ldots$	$\alpha' \wedge x_n$
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 α' can be satisfied for every individual judgment via β_1, β_2, even when α is unsatisfiable. In order to prevent α' from turning into a tautology when α is one, we added γ.

Now, we prove that $\mathcal{I} \in$ 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 α 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 α' 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 α 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 α 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 α, 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 α, we can deduce $\neg(\alpha' \wedge x_1) \in SF_q(P, \pi)$. (iii) If there are satisfying assignments for α 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 α, 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 α. 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: α is satisfiable but for its maximum satisfying assignment $x_n = 0$ holds. In the third iteration we add α' 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 α's maximum satisfying assignment. By assumption, we know that $x_n = 0$ is true in α'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: α 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 Δ_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, β_1 and β_2, but no other formula, allows us to reuse the same proof (i.e., the given order π) for the ranked agenda rule. This yields an even stricter result for the ranked agenda's winner problem's complexity, namely para-Δ_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-Δ_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 β_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 ψ_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 ψ_k^B from a set of boolean variables B' with $|B'| = nk$ in time polynomial in n, such that ψ_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 α by $\bigvee_{j\in[k]} x_i^j$ and denote the result as α_k. Then, we construct a new boolean formula $\alpha_k' = \alpha_k \wedge \psi_k^B$ and check whether α_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 $\mathcal{SK}$-Winner-Design and the $\mathcal{SK}$-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 $\mathcal{SK}$.

$\mathcal{SK}$-Winner-Design ($\mathcal{SK}$D)	
Instance:	An agenda Φ, 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 Φ_+ such that $J \subseteq \mathcal{SK}(P, \pi)$?

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

Definition 5.1. Let $\mathcal{K}$ be a complete and complement-free JA rule, Φ 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 $\mathcal{SK}(P, \pi)$ the corresponding aggregated judgment. Let $K_\pi = \mathcal{K}(P) \cap \mathcal{SK}(P, \pi)$ denote the set of formulas in the aggregated judgment also supported by $\mathcal{K}$, and $D_\pi = \mathcal{SK}(P, \pi) \setminus \mathcal{K}(P)$ 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_π 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, Φ 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 $\mathcal{SK}(P, \pi') = \mathcal{SK}(P, \pi)$.*

The intuition is, that we can rearrange every order π in such a way that all formulas supported by $\mathcal{K}$ are at the beginning of π 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 $\mathcal{SK}$-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 $\mathcal{SK}D \in$ 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_qD 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_qD 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 γ, β_1, and β_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, α is unsatisfiable. Choosing the order $\pi = ((\alpha \wedge \gamma) \vee \neg\beta_1 \vee \neg\beta_2, \beta_1, \beta_2)$ over Φ_q results in $SF_q(P, \pi) = J$. For the direction from right to left assume $\mathcal{I}$ is a no-instance and thus, α is satisfiable. Then, $F_q(P)$ is already consistent and $SF_q(P, \pi) = F_q(P) \neq J$ holds for every order π. 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 Φ, profile $P \in \mathcal{J}(\Phi)^r$ and complete and consistent judgment $J \in \mathcal{J}(\Phi)$, the corresponding $\mathcal{SK}$R-instance (Φ, 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
	$J \in \mathcal{J}(\Phi)$	$\varphi \in \Phi$	$J \in \mathcal{J}(\Phi)$	$\varphi \in \Phi$	
Δ_2^p-c., Thm. 3.1, 3.2	coNP-c., Thm. 5.3, 5.4	Σ_2^p-c., Thm. 5.7, 5.9	in P, Thm. 5.5	Π_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 Φ, 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 π 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 φ, 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 Σ_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 Σ_2^p-complete.*

Proof. Due to Theorem 5.7 it is enough to show Σ_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, ψ_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 β_j and γ. Note that by including γ, the agenda cannot contain any contradictions or tautologies. More precisely, both ψ_k^B, φ and their negations are satisfiable, even if every φ_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 π over Φ_q instead of Φ_+. 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 ψ_k^B and $m \leq k$ variables from B are satisfiable simultaneously, even if both β_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 β_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 $\bigvee_{i \in N'} \varphi_i \equiv$ TRUE holds. By contradiction, we assume $\mathcal{I}'$ to still be a YES-instance. Then, there exists an order π over Φ_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 φ must have been entailed by previously added formulas because $\varphi \notin F_q(P)$. Hence, we conclude that for the given order π it holds that $SF_q^{\sim\varphi}(P,\pi) \models (\bigvee_{i\in M} \varphi_i) \vee \gamma$, although neither γ nor any φ_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 γ is a literal, this implies that $\bigvee_{i\in M} \varphi_i \equiv$ TRUE with $|M| \leq k$ would be a solution to $\mathcal{I}$, which is a contradiction to our assumption. Therefore, such an order π cannot exist and $\mathcal{I}'$ must be a NO-instance, too. □

Again, we derive a corollary for SF_qR 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 Π_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 Σ_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 β_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-Σ_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 Φ with $|\Phi_+| = m$, 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 Φ_+ such that $|\mathcal{K}(P) \cap S\mathcal{K}(P,\pi)| \geq k$ 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_qSJ 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 π, 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 π 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 Φ 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 π over Φ_+ 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 π' where $J \cap \mathcal{K}(P)$ is at the beginning of π', 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 π'. 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 π, 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 φ 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 Π_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 Π_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 Σ_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 Σ_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 Σ_2^p Minimization Problems. In *40th Annual Symposium on Foundations of Computer Science (Cat. No. 99CB37039)*. IEEE, 465–474.

CHAPTER 8

COLLECTIVE COMBINATORIAL OPTIMISATION AS JUDGMENT AGGREGATION

In this chapter, we explore a unified model for judgment aggregation, allowing for asymmetric and/or weighted rules. We showcase how diverse collective combinatorial optimization problems can be encoded into this general model and how domain specific rules relate to known judgment aggregation rules (or their generalizations). We complement our study with complexity results on winner determination for our generalized rules.

8.1 Summary

In this work, we merge two recent generalizations of judgment aggregation into a unified framework (formally described in Subsection 2.4.4), to study collective combinatorial optimization problems as case studies. In particular, while Rey, Endriss, and de Haan [149] initiated the study on asymmetric judgment aggregation rules, Nehring and Pivato [133] considered a symmetric setting with weighted issues. We adopt and combine both approaches to model weighted, asymmetric judgment aggregation rules, which are expressive enough to capture many domain specific rules in various research fields.

As case studies, we show how multiwinner elections, participatory budgeting, collective scheduling, and collective network design (along with the novel, yet unstudied, problem of *collective placement*) can be encoded into our model. Then, we continue to relate judgment aggregation rules to domain specific optimization problems, by showing that independently formulated rules coincide with prominent judgment aggregation rules (or their asymmetric and/or weighted generalizations), if instantiated (i.e., restricted) to the respective domain.

We complement our results with a study on the computational complexity of (irresolute, existentially quantified) winner determination. In particular, although related literature for standard judgment aggregation rules is densely populated, we close remaining gaps with respect to weighted rules. For the Chamberlin-Courant rule, we show that outcome determination becomes computationally less complex in a symmetric setting with matching constraints on the in- and output. This may stimulate further research on the complexity in relaxed judgment aggregation scenarios.[52] Notably, we embed our findings into a broader context, discussing implications that arise from existing complexity gaps between judgment aggregation rules and their domain specific specializations.

8.2 Reflection on Initial Research Goals

This article fits nicely into the scope of this thesis, as we connect independently studied research fields by showing how different combinatorial optimization problems, including multiwinner elections and participatory budgeting, can be studied under a global umbrella in a unified judgment aggregation framework. As for our initial research goals, we dealt with three of the proposed research questions, introduced in Chapter 3. Most significantly, we addressed Question Q4 by *(i)* combining recent generalizations for judgment aggregation [149, 133] to model weighted, asymmetric rules and *(ii)* identifying several instances of coinciding (judgment aggregation) rules across literature for multiwinner elections, participatory budgeting, collective network design, and collective scheduling. This allowed us to extend popular frameworks into a more expressing language, allowing for additional constraints. To benefit from this generalization in an applied context, we addressed Question Q2 to determine whether the shift to the more expressive framework comes with an increase in computational complexity. Therefore, we complemented known results with a complexity study for our introduced, generalized rules. To some degree we addressed Question Q1, as our results on coinciding rules allow for a transfer of (satisfied) axiomatic results from our judgment aggregation framework to specialized, combinatorial optimization problems and vice versa (in case of violation).

[52] Let us motivate our claim with a simple additional result. De Haan [91] showed that the median rule is tractable in case the integrity constraint on in- and output is in 2-CNF. Yet, in Footnote 19 we showed that finding a satisfying assignment with a maximum hamming weight is at least as hard as finding a largest clique. Now, if the rationality constraint allows a single voter to approve all positive literals, computing the median rule must be computationally hard, as any output corresponds to a largest clique.

8.3 Publication

This work has been accepted at a special issue of the journal *Annals of Mathematics and Artificial Intelligence*, focusing on recent developments in preference handling.

[34] L. Boes, R. Colley, U. Grandi, J. Lang, and A. Novaro. "Collective Combinatorial Optimisation as Judgment Aggregation". In: *Annals of Mathematics and Artificial Intelligence* (2023)

A preliminary version of this article has undergone a review process and was accepted for presentation at the *13th Multidisciplinary Workshop on Advances in Preference Handling*.

[33] L. Boes, R. Colley, U. Grandi, J. Lang, and A. Novaro. "Collective Combinatorial Optimisation as Judgment Aggregation". In: *Proceedings of the 13th Multidisciplinary Workshop on Advances in Preference Handling*. Ed. by M. Ozturk, C. Labreuche, P. Viappiani, and S. Destercke. Vienna, Austria, 2022

Parts of this work have also been discussed in Rachael Colley's dissertation [54].

8.4 Personal Contribution

This work was initiated during my inspiring research stay at the University of Toulouse under the supervision of Umberto Grandi back in 2020. Conception, writing, and the extensive literature review was conducted jointly in equal parts together with all my co-authors Rachael Colley, Umberto Grandi, Jérôme Lang, and Arianna Novaro. Proposition 10 was contributed by me. The remaining technical results (in particular clearly marked propositions) were established in equal parts during rewarding debates, mostly with my co-author Rachael Colley, and, to a notable degree, by Arianna Novaro during the finalization of this work.

Annals of Mathematics and Artificial Intelligence
https://doi.org/10.1007/s10472-023-09910-w

Collective combinatorial optimisation as judgment aggregation

Linus Boes[1] · Rachael Colley[2] · Umberto Grandi[3] · Jérôme Lang[4] · Arianna Novaro[5]

Accepted: 29 October 2023

Abstract
In many settings, a collective decision has to be made over a set of alternatives that has a combinatorial structure: important examples are multi-winner elections, participatory budgeting, collective scheduling, and collective network design. A further common point of these settings is that agents generally submit preferences over issues (e.g., projects to be funded), each having a cost, and the goal is to find a feasible solution maximising the agents' satisfaction under problem-specific constraints. We propose the use of judgment aggregation as a unifying framework to model these situations, which we refer to as collective combinatorial optimisation problems. Despite their shared underlying structure, collective combinatorial optimisation problems have so far been studied independently. Our formulation into judgment aggregation connects them, and we identify their shared structure via five case studies of well-known collective combinatorial optimisation problems, proving how popular rules independently defined for each problem actually coincide. We also chart the computational complexity gap that may arise when using a general judgment aggregation framework instead of a specific problem-dependent model.

Rachael Colley
rachael.colley@glasgow.ac.uk

Linus Boes
linus.boes@uni-duesseldorf.de

Umberto Grandi
umberto.grandi@irit.fr

Jérôme Lang
jerome.lang@lamsade.dauphine.fr

Arianna Novaro
arianna.novaro@univ-paris1.fr

1 Heinrich-Heine-Universität Düsseldorf, Düsseldorf, Germany

2 School of Computing Science, University of Glasgow, Glasgow, Scotland

3 IRIT, Université Toulouse Capitole, Toulouse, France

4 CNRS, PSL, Paris, France

5 CES, Université Paris 1 Panthéon-Sorbonne, Paris, France

Published online: 15 December 2023

Keywords Computational social choice · Judgment aggregation · Combinatorial optimisation · Computational complexity

Mathematics Subject Classification (2010) 68R01 · 68T30

1 Introduction

Public decisions often have a combinatorial structure. Typical examples are participatory budgeting (choosing a set of projects to fund, given some budget constraints), collective scheduling (collectively deciding on the order some tasks will be executed), collective network design (collectively deciding on the edges of a network to connect multiple locations), multi-winner elections (find a committee of a fixed size), and many more—including problems which have not been given much attention, yet are worth investigating.

Many of these problems, including the four examples given above, have been studied separately while sharing many features. Their input mainly consists of individual preferences expressed *locally* (on projects, on the relative order of two projects, on edges between two nodes) rather than *globally* (on all sets of projects, all possible schedules or all possible networks).[1] The output is a feasible solution, i.e., a solution which abides by the constraints and that maximises an objective expressed via a score function. What distinguishes these problems is the nature of the constraints: the maximum budget should not be exceeded, a schedule should be a consistent ordering, a network should be a tree, etc.

The problems mentioned above are thus structurally close, and we may ask whether they could be seen as instances of a more general framework, consisting of a general language for expressing preferences and constraints, general aggregation functions, and general computational tools. If the answer is positive, each of these problems, their variants, as well as novel collective combinatorial optimisation (CCO) problems, could be expressed and solved in such a general framework without the need to study each of them separately.

We give a positive answer, and we do so by showing that *judgment aggregation* is such a suitably general framework. This framework has been initially developed to study the aggregation of binary judgments over logically interconnected issues in an agenda to gain a collective decision (cf. the survey by Endriss, [13]). Given the type of problems that we wish to study, we need a slight generalisation of standard judgment aggregation that can allow for *weighted agendas* and/or *asymmetric agendas*. We show that several aggregation rules used to solve specific problems are actually instances of existing and well-studied judgment aggregation rules, or of some of their variants (such as weighted generalisations).

However, it is worth first clarifying what our paper is not:

(i) *We do not define a brand-new framework*: most aspects of the unifying framework that we give are known from judgment aggregation; we do define weighted variants of some well-known rules that are useful to capture CCO problems.

(ii) *We do not study new problems*: instead, we restate several existing problems that have been studied independently, such that they can be analysed under a common umbrella framework. By doing so, we do however open the possibility to define and study new problems, such as what we will call *collective placement*.

[1] Since the set of solutions has a combinatorial structure, expressing preferences globally requires a high communication burden from the agents—however, using a local approach limits the types of preferences that can be expressed.

(iii) *We do not give better algorithms for specific problems*: in fact, general solvers can perform at best as well as specific solvers, and sometimes worse. Also, for several rules, the computational complexity of solving the outcome determination problem for the general case is higher than that of specific ones.

(iv) *We do not give any axiomatisation*: the axiomatisation of (judgment aggregation) rules has been investigated *per se*. The properties of the rules naturally carry over to the specific problems (although an axiomatisation of a general rule does not necessarily lead to that of its specific instantiation).

Our contribution establishes novel and fundamental connections between several lines of research and problems that have so far been studied separately. Identifying those connections may lead to meaningful insights across different areas of research. In doing so, we give an engineering flavour to judgment aggregation, a field that up until now has focused on impossibility results, axiomatisations, and computational complexity, but not yet on concrete applications to real-world problems.

Moreover, for some judgment aggregation rules the outcome can be computed via a translation into integer linear programming (ILP). Of course, such a translation could be done for each specific problem, but the existence of a general translation allows for a simpler process in comparing models, since translating a problem to judgment aggregation can be easier than translating it directly into an ILP.

There are two main streams of related literature. On the one hand, we have papers studying specific settings for collective combinatorial optimisation: these will be cited extensively when they are introduced formally in Section 3. On the other hand, we have papers adapting classical judgment aggregation to deal with weighted issues, which we discuss here below.

Rey et al. [36] provide efficient and exhaustive embeddings of participatory budgeting problems via DNNF circuits in non-weighted judgment aggregation, giving an initial axiomatic study of *asymmetric* additive rules extended from known judgment aggregation rules. Chingoma et al. [8] simulate multi-winner voting rules in judgment aggregation for both ordinal and approval-based preferences and study their complexity. Both of these works, however, do not generalise directly to other CCO settings. Nehring and Pivato [32] introduce and study a setting of judgment aggregation with weighted issues: we use their definitions to build our general framework, including the median rule of which they give an axiomatisation.

Our paper is structured as follows. In Section 2, we give an overview of judgment aggregation rules and their generalisations to weighted asymmetric agendas, and we also generalise the Chamberlin-Courant voting rule for approval ballots to judgment aggregation. In Section 3, we show how numerous collective combinatorial optimisation settings can be expressed in weighted judgment aggregation and we prove that some of the specific rules from these settings are in fact instances of judgment aggregation rules. In Section 4 we give a computational study of the rules, both from a theoretical and an experimental point of view. We conclude in Section 5.

2 Weighted asymmetric judgment aggregation

Judgment aggregation is a general framework to make collective decisions over a set of possibly interconnected issues linked by constraints. Nehring and Pivato [32] have considered a generalisation of judgment aggregation where each issue is associated with a numerical weight, while Rey et al. [36] have defined asymmetric judgment aggregation rules. We combine both approaches and consider agendas for which each item and its negation have weights (which are not required to be equal).

2.1 Formal model

A set of n agents (or voters) $\mathcal{N} = \{1, \dots, n\}$ have to take a collective decision on the acceptance of m items (projects, issues, etc.) in an agenda A. The agenda A is composed of two sets A^+ and A^-, which represent the set of positive and negative issues, respectively. Hence, $A = A^+ \cup A^-$, where $|A^+| = |A^-| = m/2$. This means that for every $a \in A^+$ there is an $\overline{a} \in A^-$ which represents the negation of a.

Each item of the agenda A has an associated weight $w_a \in \mathbb{N}_0$ that will be used in the aggregation by the rules.[2] The weight vector $\boldsymbol{w}$ collects the weights of all the m items in the agenda A. While in the most general setting, each weight can be any natural number or 0, we are also interested in some specific restrictions on the weight vectors (which determine different types of agendas) that we describe here below.

First, we can have agendas where each positive item $a \in A^+$ and its corresponding negation $\overline{a} \in A^-$ have the same weight (what we call a *symmetric* agenda, i.e., $w_a = w_{\overline{a}}$ for each $a \in A^+$; although note that each positive-negative issue pair in the agenda may have a different weight), or agendas where the negated item has weight zero (what we call an *asymmetric* agenda, i.e., $w_{\overline{a}} = 0$ for each $\overline{a} \in A^-$). Second, we can have agendas where the items' weights are in the set $\{0, 1\}$ (what we call a *binary* agenda) or where they are in $\mathbb{N}_0$ (what we call a *weighted* agenda).

By combining the above cases, we derive the following four natural variants of judgment aggregation, defined by the corresponding restrictions on the weight vector of the agenda, which we will focus on in the rest of this paper. Note that an agenda may be neither symmetric nor asymmetric (i.e., $w_a, w_{\overline{a}} \in \mathbb{N}$, yet $w_a \neq w_{\overline{a}}$).

Standard judgment aggregation ($w_a = w_{\overline{a}} = 1$) In standard judgment aggregation, denoted by $\boldsymbol{w}^{sym}_{bin}$, all issues (and their negations) have equal weights. Formally, for any $a, a' \in A$ we have that $w_a = w_{a'}$. Without loss of generality, we can assume that all weights are equal to 1, thus yielding a *symmetric binary* agenda.

Asymmetric judgment aggregation ($w_a = 1$; $w_{\overline{a}} = 0$) For *binary asymmetric* agendas, which were studied by Rey et al. [36], and which we denote by $\boldsymbol{w}^{asym}_{bin}$, we assume that $w_a = 1$ for all $a \in A^+$ and $w_{\overline{a}} = 0$ for all $\overline{a} \in A^-$. Intuitively, when an item's weight is 0, the support for this item will be discarded when computing an outcome. Thus, in the setting by Rey et al. [36], asymmetric weights allow for rules to be biased towards the acceptance of positive agenda items.

Weighted judgment aggregation ($w_a = w_{\overline{a}} \in \mathbb{N}_0$) In *symmetric weighted* judgment aggregation, which we denote by $\boldsymbol{w}^{sym}_{we}$, different items of the agenda can have different weights, yet an item and its dual must have the same weight, $w_a = w_{\overline{a}} \in \mathbb{N}_0$. This was described by Nehring and Pivato [32].

Weighted asymmetric judgment aggregation ($w_a \in \mathbb{N}_0$; $w_{\overline{a}} = 0$) Finally, in the $\boldsymbol{w}^{asym}_{we}$ restriction we have $w_{\overline{a}} = 0$ for all $\overline{a} \in A^-$ and $w_a \in \mathbb{N}_0$ for all $a \in A^+$.

In addition to being able to capture standard judgment aggregation, the setting we provide is more general in two ways: the presence of weights and the possible asymmetry between

[2] Although Nehring and Pivato [32] assume real-valued weights, integer weights allow us to use compact languages for the constraints; a further generalisation to values in $\mathbb{R}$ (or to negative values) for the weights is also possible.

Springer

issues and their negations. The latter generalisation goes slightly in the direction of belief merging (see [28]); however, we assume that all agents report judgments on the same agenda, unlike belief merging.[3]

For an agenda A and an agent $i \in \mathcal{N}$, an agent's ballot is a vector $B_i \in \{0, 1\}^m$, where each entry represents the agent's decision on a fixed agenda item. In this context, the notation $B_i(a)$ refers to the entry of vector $B_i \in \{0, 1\}^m$ for issue $a \in A$. The collection of the agents' ballots is a profile $\boldsymbol{B} = (B_1, \ldots, B_n)$.

Constraints can be imposed on a weighted judgment aggregation problem, either on the collective outcome (e.g., abiding by a budget constraint) or on the individual ballots (e.g., approving a minimal number of items). Following Endriss [14] we call the former *feasibility* constraints (denoted by Γ_F) and the latter *rationality* constraints (denoted by Γ_R). Throughout the paper, we will express these constraints as sets of linear (in)equalities, allowing the constraints from many CCO settings that include numerical values to be formalised clearly.[4]

Moreover, $\mathcal{B}_R \subseteq \{0, 1\}^m$ is the set of all ballots satisfying the rationality constraints, while $\mathcal{B}_F \subseteq \{0, 1\}^m$ is the set of outcomes satisfying the feasibility constraints. Since agendas include an issue and its negation, we always assume that all ballots and all outcomes accept either an issue or its negation (no matter the issues' weights). Additionally, when imposing rationality and feasibility constraints, we require that any voter's ballot $B_i \in \{0, 1, \}^m$ must be rational, i.e., $B_i \in \mathcal{B}_R$, and any outcome of an aggregation method $X \in \{0, 1\}^m$ must be feasible, i.e., $X \in \mathcal{B}_F$.

An important remark to make here is that the weights will be used by our rules as a proxy for the agents' satisfaction during the aggregation step in order to identify the optimal outcomes. However, as we shall see for the case of participatory budgeting (in Section 3.2), in some CCO settings the weights will also determine the feasibility of the outcomes themselves: i.e., we could have a participatory budgeting scenario where the weights associated to projects correspond to their costs (thus, the satisfaction of an agent is the total cost of the accepted items they approve of), but they also appear in the constraints (i.e., the budget limit should not be exceeded).

In the following, we sometimes use $j \in [x, y]$ as a shorthand for $j \in \{x, \ldots, y\}$, and $j \in [x]$ as a shorthand for $j \in \{1, \ldots, x\}$.

2.2 Weighted asymmetric judgment aggregation rules

In this section we recap some well-studied rules from judgment aggregation, but we define them generally in the sense that the agenda items can have any weight vector (recall that, with a slight abuse of terminology, we refer to the *weighted* versions of the rules when the weights are not binary).

We call a *weighted asymmetric judgment aggregation rule* a function F that takes as input a rational profile $\boldsymbol{B} \in \mathcal{B}_R$, set of feasibility constraints Γ_F, and weight vector $\boldsymbol{w}$ for the items in an agenda A and it gives as output a set of feasible outcomes (i.e., where $X \in \mathcal{B}_F$ for each outcome X). Observe that rules are thus *irresolute*, in the sense that they may return a set of tied feasible outcomes.

[3] See [19] for a comparison of belief merging and judgment aggregation.

[4] Note that the constraints can be expressed in many ways: for instance, in the standard (symmetric) framework of binary judgment aggregation, they are usually expressed as formulas of propositional logic [23].

2.2.1 The (weighted) median rule

The *median rule* finds the outcomes that globally minimise the number of changes between the profile and the outcome or equivalently, maximise the number of agreements between the agents' ballots and the outcome. The *weighted median rule* extends the median rule by maximising the total sum of the *weights* of accepted items in the outcome and each of the agents' ballots.

Definition 1 The *(weighted) median rule* takes a rational profile $\boldsymbol{B}$, feasibility constraint Γ_F, and agenda weights $\boldsymbol{w}$ and gives a set of outcomes found by:

$$\underset{\{X \mid X \in \mathcal{B}_F\}}{\arg\max} \sum_{i \in \mathcal{N}} \sum_{a_j \in A} w_{a_j} \times B_i(a_j) \times X(a_j).$$

We may choose any weight vector paired with this rule, however, when paired with specific weight vectors, the function aligns with specific rules. When the agenda's weight vector is restricted to be unary (i.e., symmetric and binary, $\boldsymbol{w}_{bin}^{sym}$), this rule reflects the standard median rule (which we will refer to as *Med*); while under symmetric non-binary weights, $\boldsymbol{w}_{we}^{sym}$, we obtain the symmetric weighted median rule defined byNehring and Pivato [32] (which we will refer to as *WMed*).

Observe that *Med* and *WMed* differ in name, yet they are the same function with different weights being used for the aggregation. In the former, the weights of the items, for aggregation purposes, are binary. Hence, the satisfaction of the agents is based on the cardinality of the intersection between the outcome and voters' ballots only. Note that this does not stop the feasibility constraint to use the weights of the items when determining if an outcome is feasible. In the latter, i.e., *WMed*, the issues' weights represent the satisfaction of the voters: instead of getting one point for every item approved by both the outcome and the voters' ballots, we now get the sum of the items' weights which are approved by both.

2.2.2 The (weighted) egalitarian rule

The standard egalitarian rule outputs the outcomes that maximise the minimum number of projects approved by any agent in the outcome. When the weight vector is restricted to $\boldsymbol{w}_{bin}^{sym}$ it is called the d_H-max rule by Lang et al. [31] and the *MaxHam* rule by Botan et al. [4].

Definition 2 The (weighted) *egalitarian rule* takes a rational profile $\boldsymbol{B}$, feasibility constraint Γ_F, and agenda weights $\boldsymbol{w}$ and gives a set of outcomes found by:

$$\underset{\{X \mid X \in \mathcal{B}_F\}}{\arg\max} \min_{i \in \mathcal{N}} \sum_{a_j \in A} w_{a_j} \times B_i(a_j) \times X(a_j).$$

In general, we will refer to this rule as *Egal* when weights are binary and as *WEgal* when weights are non-binary ($\boldsymbol{w}_{we}^{asym}$ or $\boldsymbol{w}_{we}^{sym}$). Although egalitarian rules have also been studied in belief merging (see, e.g., [18]), to the best of our knowledge *WEgal* is new. Its motivation is natural in participatory budgeting instances with few agents, where we may want to ensure that each agent agrees with the funded projects to some minimum level.

2.2.3 The (weighted) ranked agenda rule

The ranked agenda rule was studied by Lang and Slavkovik [30] and its *leximax* refinement was studied by Nehring et al. [33]. The rule iteratively considers issues following the order induced by the support received, rejecting an issue if its addition would break the feasibility constraint. There are two variants of the rule, depending on whether ties are broken immediately, using a tie-breaking rule, or if all tie-breaking possibilities are considered in parallel. For simplicity, we only consider the former variant.

Algorithm 1 The (weighted) ranked agenda rule.

```
1: Input: Γ_F, B, A, w
2:   Θ := {Γ_F} and X := {0}^|A|
3: Order the issues of A w.r.t. ∑_{i∈N} B_i(a) × w_a in descending order
4: for each issue a_j in the ordering do
5:     if Θ and a_j is consistent then
6:         X(a_j) := 1 and Θ := Θ ∪ {a_j}
7:     end if
8: end for
9: return X
```

The rule follows Algorithm 1. It has as input the constraint Γ_F, the profile of ballots $\boldsymbol{B}$, the agenda A, and the weight vector $\boldsymbol{w}$; the output is an outcome $X \in \mathcal{B}_F$. On line 2, Θ is initialised to be a set containing Γ_F and the outcome vector X sets a 0 for each item. It then orders the items in A with respect to their weighted support (using a linear tie-breaking rule when items have equal support). Following this order of items, in the for-loop on line 4, it first checks if the addition of this item breaks feasibility given the currently accepted items. If feasibility is respected, then the outcome for that item is set to 1 and the item is added to Θ (otherwise, its negation will be in the outcome). This algorithm can be altered to get other ranked rules, such as the greedy Chamberlin-Courant rule (see [41]).

For a non-binary agenda, we will refer to this rule as *WRank* and for a binary agenda we will refer to it as *Rank*.

2.2.4 The Chamberlin-Courant rule

The *Chamberlin-Courant voting rule* (CC) was originally introduced for ordinal preferences by Chamberlin and Courant [7] as a way to try to ensure that all voices are present in deliberation. We consider a variant for approval-based preferences, studied by Skowron and Faliszewski [37] for multi-winner elections and generalised to participatory budgeting by Talmon and Faliszewski [41].[5] For approval-based preferences, an outcome should maximise the number of agents who have at least one item in the outcome that they approve of. We define the rule here for general agendas: we say that an agent is *satisfied* by an outcome if there is at least one issue with non-zero weight approved by the agent and contained in the outcome; then, the rule outputs every outcome that maximises the number of satisfied agents.

Definition 3 The *Chamberlin-Courant* rule takes a rational profile $\boldsymbol{B}$, feasibility constraint Γ_F, and agenda weights $\boldsymbol{w}$ and gives a set of outcomes found by:

$$CC(\boldsymbol{B}, \Gamma_F, \boldsymbol{w}) = \underset{\{X \mid X \in \mathcal{B}_F\}}{\arg\max} \sum_{i \in \mathcal{N}} \min(1, \sum_{a_j \in A} X(a_j) \times B_i(a_j) \times w_{a_j}).$$

[5] In a related approach, Chingoma et al. [8] generalised the Chamberlin-Courant rule and the proportional approval voting rule (PAV) to judgment aggregation, relying on weak rankings to model either dichotomous or ordinal preferences.

First, note that changing the weight of an issue from some non-zero number to another non-zero number has no impact on the outcome: only changing non-zero weights to zero and vice-versa can affect the outcome (without loss of generality non-zero weights can be 1). Second, the rule makes particular sense for asymmetric agendas, such as those for participatory budgeting or multi-winner elections.

2.3 Examples

We now provide an example illustrating the four rules we just introduced, both in their weighted and binary versions.

Let $\mathcal{N}$ be a set of 8 voters and $A^+ = \{i, j, k, \ell, m, n\}$ be a set of 6 (positive) issues. Each positive issue has an associated cost, modelled by vector $c = (1, 2, 2, 2, 3, 4)$. In this example, costs are used to determine if an outcome is feasible. Moreover, when the weights are non-binary, they will correspond to the issues' costs. We will only consider an asymmetric agenda, thus, the weight vectors will be restricted to be of the form $\boldsymbol{w}_{bin}^{asym}$ or $\boldsymbol{w}_{we}^{asym}$. Hence, for each $\overline{a} \in A^-$, $w_{\overline{a}} = 0$.

The feasibility constraint states that the total cost of the collectively accepted issues is no greater than 5. We first study the profile $\boldsymbol{B}$ given in Table 1. The line labelled *sum* gives the total support for each of the issues in the profile, while the line labelled *weighted sum* gives the total support for that issue times its weight.

Binary asymmetric weights We first consider the aggregation weights to be binary and asymmetric. Thus, the aggregation weight vector $\boldsymbol{w}$ is such that $w_a = 1$ for $a \in A^+$ and $w_a = 0$, otherwise. Given $\boldsymbol{w}$, we compute the rules *Med*, *CC*, and *Rank*: each of them considers the amount of support each item of A^+ has received, as per the *sum* line in the table.

The unique outcome of *Med* is accepting only i, j, and k as it has the highest total support of 13 and all other feasible outcomes have smaller total support.

The *Rank* rule returns the outcome that accepts only i and m. It orders the issues with respect to how much support they have received: i.e., i, m, j, k, n, ℓ (breaking ties alphabetically); then, it accepts i and then m as their total cost is 4. None of the remaining issues can be added without exceeding the budget limit.

The *CC* rule returns the outcome that only accepts ℓ and m as this is the only feasible outcome in which all voters approve of at least one of the issues.

Weighted asymmetric weights We now consider a weighted asymmetric vector $\boldsymbol{w}'$, where $w'_a = c(a)$ for all $a \in A^+$, giving $\boldsymbol{w}'^+ = (1, 2, 2, 2, 3, 4)$, i.e., the weight of each issue is its cost, and each issue in A^- has a weight of 0. The outcomes of the rules *WMed* and *WRank* on $\boldsymbol{w}'$ are given in Table 1 (we do not consider the *CC* rule on non-binary weights).

The rule *WMed* returns two tied outcomes that maximise the total weighted support. The first outcome accepts only issues j, and m, while the second accepts only issues k and m: both have a total weighted support of 23.

The rule *WRank* first orders the issues with respect to their weighted support (the numbers in the *weighted sum* line), giving the order: n, m, j, k, ℓ, i (ties are broken alphabetically). Then, it accepts issue n with cost 4. It then must reject m, j, k, and then ℓ, in this order, as their acceptance would exceed the limit of 5 given by the constraint. Then issue i is added, giving the outcome where only n and i are accepted.

Egalitarian rules For the egalitarian rules, we consider a different profile $\boldsymbol{B}'$ given in Table 2 with $\mathcal{N} = [6]$, where we consider the same issues as before except j, i.e., $A^+ \setminus \{j\}$. Our weight

Table 1 Profile $\boldsymbol{B}$ from the example given in Section 2.3

	i	j	k	ℓ	m	n
cost	1	2	2	2	3	4
voter 1	1	0	1	0	1	0
voter 2	0	1	1	1	0	1
voter 3	1	1	1	0	1	1
voter 4	1	1	0	0	1	1
voter 5	1	1	0	1	0	1
voter 6	1	0	1	0	1	0
voter 7	0	0	0	1	0	0
voter 8	0	0	0	0	1	0
sum	5	4	4	3	5	4
Med	1	1	1	0	0	0
CC	0	0	0	1	1	0
Rank	1	0	0	0	1	0
weighted sum	5	8	8	6	15	16
WMed	0	1	0	0	1	0
	0	0	1	0	1	0
WRank	1	0	0	0	0	1

We give the total number of approvals for each issue (*sum*), the weighted number of approvals (*weighted sum*), and the outcome of the different rules (including possible ties, as seen for *WMed*)

vectors remain the same excluding the entry corresponding to j. The cost of each issue is given in $c' = (1, 2, 2, 3, 4)$, and the budget constraint remains at 5.

First consider *Egal*, where the weight vector has $w_a = 1$ if $a \in A^+\backslash\{j\}$, and $w_a = 0$ if $a \in A^-\backslash\{\bar{j}\}$. In this instance, the outcomes of *Egal* contain, for each voter, at least one issue

Table 2 Profile $\boldsymbol{B}'$ from Section 2.3 used to show the outcome(s) of *Egal* and *WEgal*

	i	k	ℓ	m	n
cost	1	2	2	3	4
voter 1	1	1	0	1	0
voter 2	0	1	1	0	0
voter 3	0	1	0	1	1
voter 4	1	0	0	1	1
voter 5	1	0	1	0	1
voter 6	1	1	0	1	0
Egal	1	1	1	0	0
	1	1	0	0	0
	0	0	1	1	0
WEgal	0	0	1	1	0

that they approve of. In particular, each of the 6 voters approves of either issue ℓ or m, so *Egal* returns the outcome approving these two issues (no more items can be added without exceeding the budget limit). Then, there is only one feasible outcome accepting three issues from the positive agenda: i, k and ℓ. Here the least satisfied voters (voters 3 and 4) approve exactly one issue in the outcome. Note that the outcome accepting only issues i and k is returned by *Egal*, as the least satisfied voter (voter 3) also approves of only one issue.

As *WEgal* is a weighted judgment aggregation rule (i.e., using a non-binary weight vector), we let the issues' weights be their cost. Thus, $w_a = c(a)$ for $a \in A^+ \backslash \{j\}$ and $w_a = 0$ if $a \in A^- \backslash \{\overline{j}\}$. Then, only the outcome accepting issues ℓ and m is returned by *WEgal*, as every agent gets at least a minimum weight of 2 (out of 5) that they approve of.

3 Collective combinatorial optimisation: five problems

In this section, we present five examples of collective combinatorial optimisation (CCO) problems from the literature, and we model them in judgment aggregation. A CCO problem has the following characteristics. First, a collective decision must be taken to decide which discrete items should be accepted from a given finite set. In many examples of CCO settings, each of the items will have a cost, which may be used by the constraints. In general, the combinatorial nature of the problem then arises from the presence of constraints specifying what is a feasible outcome. CCO rules will then take a rational profile of ballots and return a feasible collective combinatorial outcome, optimising some metric of satisfaction. Note that rationality and feasibility constraints here correspond to the specifics of the CCO setting in question.

We show that specific rules, studied independently within each CCO setting, are instances of the rules defined in Section 2.2. Formally, a CCO rule $\mathcal{R}$ *is an instance of* a judgment aggregation rule $\mathcal{R}'$ if there is a translation from any profile $\boldsymbol{B}$ of this specific CCO setting into a judgment aggregation profile $\boldsymbol{B}'$ such that the outcome of $\mathcal{R}$ on $\boldsymbol{B}$ is equivalent to the outcome of $\mathcal{R}'$ on $\boldsymbol{B}'$, i.e., they correspond to the same collective decision.

3.1 Multi-winner elections

This well-studied framework models the collective selection problem of a set of candidates. The candidates have equal weight, which can be assumed to be unitary without a loss of generality. The agenda is $A = \{a, \overline{a} \mid a \text{ is a candidate}\}$. Furthermore, we mainly consider CCO rules in which the agents only vote on the acceptance of the candidates, the exception being the *minimax* approval rule (details are given in the following). Thus, the agenda is binary and asymmetric, i.e., for each pair $a, \overline{a} \in A$, $w_a = 1$ and $w_{\overline{a}} = 0$. We thus only consider the judgment aggregation rules with binary weights. The feasibility constraint requires that exactly a given number $k \in \mathbb{N}$ of candidates are elected, thus $\mathcal{B}_F = \{X \mid \sum_{a \in A} X(a) \times w_a = k\}$. The ballots represent the agents' approval of the candidates: they can approve as many candidates as they like, or exactly k candidates ($\mathcal{B}_R = \mathcal{B}_F$).

The following proposition shows the correspondence between rules in the literature on multi-winner voting and our general judgment aggregation rules. Definitions of multiwinner voting rules can be found in the work of Lackner and Skowron [29].

Proposition 1 *The following multi-winner rules are instances of their judgment aggregation counterparts:*

i) the standard multi-winner approval voting rule, that outputs the most approved k candidates, is an instance of both Med and Rank (modulo tie-breaking);
ii) the Chamberlin-Courant rule for approval ballots (see [37]) is an instance of CC;
iii) the minimax approval voting rule by Brams et al. [5], that outputs the outcomes minimising the maximum, over all agents i, of the Hamming distance between the outcome and i's ballot, is an instance of Egal with binary symmetric weights ($\mathbf{w}_{bin}^{sym}$).[6]

Proof For asymmetric agendas equipped with a multi-winner constraint, the median rule *Med* selects all subsets of candidates that maximise the (sum of the) overlap of voters' approved issues with the outcome. This is exactly what the *multi-winner approval voting rule* does. The ranked agenda rule *Rank* sorts the issues by voters' support first and then sequentially adds the best k issues to the outcome. Considering parallel-universe tie-breaking, we derive exactly those outcomes that are output by the median rule.

As the *Chamberlin-Courant rule for approval ballots* was adapted from Skowron and Faliszewski [37] into our judgment aggregation framework, the instantiation follows by design. Both in multi-winner elections and judgment aggregation (equipped with a multi-winner constraint), the *Chamberlin-Courant rule* selects those fixed-size subsets of issues that maximise the number of voters that approve of at least one (positive) issue.

The *minimax approval voting rule* by Brams et al. [5] selects those outcomes that minimise the maximum Hamming distance to the voters' approval ballots. Botan et al. [4] point out that their judgment aggregation rule *MaxHam* generalises the minimax approval voting rule. Note that maximising the minimum support instead of minimising the maximum lack of support is only a difference in modelling. In particular, as we consider symmetric agendas, each agent supports exactly half of the issues (counting rejections). Thus, the Hamming distance between a voter's ballot and an outcome can be derived from counting the supported issues, subtracting $m/2$, and multiplying by -1 (and swapping minimisation and maximisation operators due to the sign change). □

The egalitarian multi-winner rule appears to be novel—although a recent paper on participatory budgeting proposed an asymmetric egalitarian rule [40]. The rule outputs the committees of k candidates that maximise the minimum number of committee members approved by any agent. It is close to the rules studied by Aziz et al. [2], who however consider ballots to be rankings over alternatives.

3.2 Participatory budgeting

Participatory budgeting (PB) is a class of collective selection problems, generalising multi-winner elections, where the agents approve projects to be funded by a limited resource (e.g., a monetary budget). A PB problem consists of a set of projects P, and each $p \in P$ has a cost c_p if implemented. The PB rule will return a set of selected projects with a total cost that must not exceed the budget limit $\ell \in \mathbb{N}$.

We will now rephrase the PB problem in terms of our judgment aggregation notation introduced in Section 2.1. The selection agenda A contains issues for each project $p \in P$ represented by a_p. The agenda has asymmetric weights, i.e., $w_{\overline{a}} = 0$ for every project $a \in A$. We focus on the case where the agenda weights are either binary ($w_p = 1$ for each $a_p \in A^+$) or a weighted agenda (where $w_p = c_p$), depending on the notion of satisfaction required by the rule.

[6] The link between minimax approval voting and judgment aggregation was discussed by Grossi and Pigozzi [24].

Note that in participatory budgeting, the feasibility of the outcome is always determined by the projects' costs. The weights, as mentioned in Section 2, are instead used in the aggregation to quantify the voters' satisfaction.

Regarding the feasibility of the outcomes, the cost of the collectively selected projects must not exceed the budget limit $\ell \in \mathbb{N}$. Hence,

$$\mathcal{B}_F = \{X \mid \sum_{a \in A^+} X(a) \times c_a \leq \ell\}$$

are the feasible outcomes, with A^+ the positive issues of A.

If there are no rationality constraints, $\mathcal{B}_R = \{0, 1\}^m$, agents can approve any number of items (regardless of their costs); if $\mathcal{B}_R = \mathcal{B}_F$, agents must submit their ideal allocation under the budget limit. Related problems include collective knapsack or knapsack voting [22], where rationality and feasibility constraints coincide, and weighted committee selection [27].

Proposition 2 *The following PB rules are instances of their judgment aggregation counterparts:*

$i)$ *the max rule with cardinality satisfaction from Talmon and Faliszewski [41] is an instance of Med;*
$ii)$ *the generalised approval-based Chamberlin-Courant rule by Talmon and Faliszewski [41] is an instance of CC;*
$iii)$ *the max rule with cost satisfaction from Talmon and Faliszewski [41] is an instance of WMed;*
$iv)$ *the greedy rule with cardinality satisfaction by Talmon and Faliszewski [41] is an instance of Rank;*
$v)$ *the greedy rule with cost satisfaction from Talmon and Faliszewski [41] is an instance of WRank;*
$vi)$ *the maxmin participatory budgeting rule from Sreedurga et al. [40] is an instance of WEgal;*
$vii)$ *the individually best knapsack rule from Fluschnik et al. [20] is an instance of Med when their utilities are binary;*
$viii)$ *the diverse knapsack rule from Fluschnik et al. [20] is an instance of CC when their utilities are binary.*

Proof We begin with the rules by Talmon and Faliszewski [41]. The *max rule with cardinality satisfaction* and the *generalised approval-based Chamberlin-Courant* respectively generalise the multi-winner rules *approval voting rule* and *Chamberlin-Courant rule for approval ballots*, with the only difference that feasibility is determined by a participatory budgeting constraint. Following Proposition 1, the rules are instances of *Med* and *CC*.

In participatory budgeting, a voter's cost-based satisfaction corresponds to the funds spent on projects the voter approves of. The *max rule with cost satisfaction* selects those feasible bundles that maximise the (sum of) voters' cost-based satisfaction. This is exactly what *WMed* for weighted asymmetric agendas does when the weights model the respective projects' costs.

The *greedy rule with cardinality (resp. cost) satisfaction* constructs an outcome sequentially. The issues are first ranked either by the sum of cardinality satisfaction (i.e., the number of supporting voters) or the sum of cost satisfaction, then projects are selected by descending total satisfaction (with some tie-breaking rule), skipping projects that would break the feasibility constraint. Translated to judgment aggregation with asymmetric agendas, the voters' satisfaction with each issue is preserved. Assuming the same tie-breaking scheme is used, (W)*Rank* ranks the projects in the same way as the *greedy rule with cardinality (cost)* *satisfaction* and issues are added to the outcome sequentially, skipping issues that break feasibility.

The *maxmin participatory budgeting rule* from Sreedurga et al. [40] also assumes cost-based voters' satisfaction. The rule selects those outcomes that maximise satisfaction for the least satisfied voter, analogously to *WEgal*.

For the remaining two rules by Fluschnik et al. [20], the voters assign each item a non-negative integer utility. Assuming these utilities are binary, each item has an individual voter's utility of zero or one. Given a participatory budgeting constraint, the *individually best knapsack rule* selects a subset of items that maximises the (sum of the) voters' utilities in an analogous way to *Med*. In contrast, the *diverse knapsack rule* maximises the number of voters that have a non-zero utility (for non-binary utilities every voter is represented by the utility of her most preferred item in the outcome). For binary utilities, this corresponds to *CC*. □

3.3 Collective networking

In the problem of collective networking, the agents have to design a common network—whether the network consists of water pipelines, internet services, or travel connections between countries. The agents specify which links they approve of, and the goal is to find a spanning tree from such input, i.e., an undirected acyclic graph that includes all nodes, maximising the satisfaction of the agents. This problem has been introduced and studied by Darmann et al. [10, 11].

Given an undirected network $G = (V, E)$ a networking agenda is the set of items $A = \{a_{ij}, \overline{a_{ij}} \mid (i, j) \in E\}$, where $w_{\overline{a_{ij}}} = 0$ for all $\overline{a_{ij}} \in A^-$ (i.e., the agenda has asymmetric weights). Then, c_{ij} is the cost of adding edge (i, j) to the outcome network. Darmann et al. [11] consider edges with costs but no budget limit determining what is a feasible outcome, as they assume that some central authority will fund any outcome. As for participatory budgeting, we can consider either $w_{a_{ij}} = c_{ij}$ or $w_{a_{ij}} = 1$, depending on how the rule is going to model the agents' satisfaction for building such a connection.

The set of accepted edges must form a spanning tree (i.e., acyclic and connected tree), this is reflected in the feasibility constraints—and a budget limit can also be imposed. These constraints can be formulated as linear inequalities in many ways.[7] We here focus on the single commodity flow model by Abdelmaguid [1], where we first move from undirected to directed graphs, and we then forget the direction of the edges to obtain the collective spanning tree. We have $|E|$ variables a_{ij} stating whether (i, j) is in the collective spanning tree, and $2|E|$ variables y_{ij} and y_{ji} in set Y for the two directions of each edge in E. Each $y_{ij} \in \mathbb{N}$ describes the flow going from node i to node j.

We also have $|V|$ constraints as follows, for $j \in V$:

$$\sum_{i:(i,j)\in E} (y_{ij} - y_{ji}) = \begin{cases} 1 - |V|, & \text{if } j = 1 \\ 1, & \text{otherwise} \end{cases} \tag{1}$$

The first case accounts for the (artificial) root of the tree $j = 1$, having no in-flowing edges. Thus, $y_{i1} = 0$ for all $(i, 1) \in E$ and the out-flowing edges have a total weight of $|V| - 1$. The second case ensures that in a spanning tree, the in-flowing weight exceeds the out-flowing by one.

[7] Note that without the use of linear inequalities, constraints can still be expressed compactly (only adding a polynomial number of variables) when weights are unary (see [25]).

Next, we ensure that directed edges correspond to the undirected edges:

$$y_{ij} \leq (|V| - 1)x_{ij} \quad \text{and} \quad y_{ji} \leq (|V| - 1)x_{ij}. \tag{2}$$

For every $(i, j) \in E$, the constraints impose that each of y_{ij} and y_{ji} can carry flow only when x_{ij} is in the spanning tree. Finally, the tree must have $|V| - 1$ edges:

$$\sum_{(i,j)\in E} x_{ij} = |V| - 1. \tag{3}$$

Example 1 Four cities want to improve their train connections (illustrated in Fig. 1) and need to decide which rails should become high-speed. Let $G = (V, E)$ represent the cities and candidate rails, where $V = \{h, i, j, k\}$ and $E = \{(h, i), (h, k), (i, j), (i, k), (j, k)\}$. Thus, the agenda is $A^+ = \{a_{hi}, a_{hk}, a_{ij}, a_{ik}, a_{jk}\}$, and $(1, 2, 4, 3, 2)$ are the respective costs to build such connections, in millions of euros. Assuming the cities themselves may submit a preference on which connections to upgrade, h may want a faster connection with cities i and k, where many of its citizens work, thus submitting ballot $B_h = (1, 1, 0, 0, 0)$. A solution would then need to choose which connections to improve, ensuring that all cities are connected by high-speed rails (possibly abiding by a budget if previously specified).

We now show that the specific rules introduced for the collective network problem are all instances of judgment aggregation rules.

Proposition 3 *The following collective network rules are instances of their judgment aggregation counterparts:*

i) the maximum collective spanning tree from Darmann et al. [10] is an instance of Med;
ii) the maximin voter satisfaction problem for approval voting for spanning trees from Darmann et al. [11] is an instance of Egal;
iii) the greedy algorithm for the maximum spanning tree problem from Escoffier et al. [17] is an instance of Rank, when restricted to approval ballots.

Proof We now show that the procedure to find the maximum collective spanning tree from Darmann et al. [10] is an instance of *Med*. Finding the maximum collective spanning tree equates to finding the spanning tree that maximises the total support. Support here is determined by the number of agents who vote on an edge in the initial graph which is included in the spanning tree. We see that *Med* gives the same solution, as it returns a feasible outcome (in the case of spanning trees, with respect to (1), (2) and (3)) such that the total support of the issues (i.e., total support on the edges) is maximised. Thus, the maximum collective spanning tree from Darmann et al. [10] is an instance of *Med*.

We then show that the maximin voter satisfaction problem for approval voting as defined by Darmann et al. [11] is an instance of *Egal*. Its scoring function for approval voting assigns

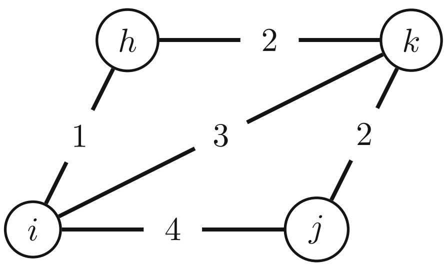


Fig. 1 Graph given in Example 1. Each node represents a city, each edge is an existing train connection, and an edge's weight is the cost to upgrade a high-speed connection

a point for each edge a voter approves of in a given spanning tree. Their egalitarian operator finds the spanning trees which maximise the minimum approval score of any agent. This coincides with the formula given in Definition 2 with binary asymmetric weights.

We next show that the greedy algorithm for the maximum spanning tree problem from Escoffier et al. [17] is an instance of *Rank*, when restricted to approval ballots. The algorithm provided by Escoffier et al. [17] orders the items by their approval level by the scoring function used. The items are added in this order only if they do not break feasibility. When the setting is restricted to approval voting, i.e., the voters can give only a valuation of 1 or 0 to every edge, this corresponds to the number of agents in support of an edge. Therefore, given a profile of votes, both the maximum spanning tree problem from Escoffier et al. [17] and *Rank* create the same ordering of the items (provided that they follow the same tie-breaking rule). As in Algorithm 1, the items are added with respect to this ordering and are only rejected when their addition would entail that the resulting graph is not a spanning tree (as per (1), (2) and (3)). Hence, the greedy algorithm for the maximum spanning tree problem from Escoffier et al. [17] is an instance of *Rank*. □

3.4 Collective scheduling

Let $P = \{p_1, \ldots, p_m\}$ be a set of (at least two) jobs to be performed on a single machine, with execution time t_x for job $p_x \in P$. The agents submit transitive and asymmetric orderings over P, indicating their preferred order of execution of the jobs, which is then decided by a collective rule. Pascual et al. [35] assume that the output schedule has no gaps and is complete (hence, $\mathcal{B}_R = \mathcal{B}_F$): the setting is thus equivalent to the aggregation of orderings of alternatives, where the alternative can have different durations (similarly to costs for participatory budgeting).[8]

Let A with $A^+ = \{a_{x \prec y}, a_{y \prec x} \mid p_x, p_y \in P\}$ be a scheduling agenda, where $a_{x \prec y}$ being accepts represents the support of p_x being scheduled before p_y, whereas $a_{y \prec x}$ represents support of p_y being scheduled before p_x. For an agent i, who provides a complete ranking over the projects, it holds that $B_i(a_{y \prec x}) = 1 - B_i(a_{x \prec y})$, and similarly for their negations. The agenda items are either weighted or binary and are usually asymmetric. We focus here on the binary asymmetric setting, where the weights of all of the items in A^+ are set to 1, and those in A^- are 0. In the weighted setting, many different asymmetric weight vectors could be considered, e.g., weights corresponding to the jobs' durations (see the discussion at the end of this section).

Agents submit complete rankings, so their ballots must approve exactly half of the positive agenda items in A^+, i.e., either $a_{x \prec y}$ or $a_{y \prec x}$ for all $\{p_x, p_y\} \subseteq P$. For the full agenda A, each agent approves half of the items (i.e., an item or its negation), even in case we want to model a voter submitting an incomplete schedule.

The outcome X of the collective scheduling problem must be a linear order of the jobs: thus, the feasibility constraints must impose transitivity and asymmetry of scheduled jobs. These can be easily formulated as linear inequalities.

Example 2 A faculty is scheduling the mandatory courses $P = \{p_1, p_2, p_3, p_4\}$ for the first-year students, and the faculty members have to decide on their ordering. Then the positive agenda is given by $A^+ = \{a_{1\prec 2}, a_{2\prec 1}, a_{1\prec 3}, a_{3\prec 1}, a_{1\prec 4}, \ldots\}$. Professor i thinks that p_2 should come first, then p_1 second, p_3 third, and that p_4 should come last. Hence, i approves of all

[8] The connection between the aggregation of (preference) orderings over alternatives without durations and judgment aggregation is well-known, and thus known how the rules in both settings correspond to each other Endriss [13].

items in $\{a_{2\prec x}, a_{\overline{x\prec 2}} \mid p_x \in \{p_1, p_3, p_4\}\} \cup \{a_{1\prec x}, a_{\overline{x\prec 1}} \mid p_x \in \{p_3, p_4\}\} \cup \{a_{3\prec 4}, a_{\overline{4\prec 3}}\}$, and reject the remaining items.

Proposition 4 *The following collective scheduling rules are instances of their judgment aggregation counterparts:*

i) the utilitarian aggregation rule with swap distance from Pascual et al. [35] is an instance of Med;
ii) the egalitarian aggregation rule with swap distance from Pascual et al. [35] is an instance of Egal.

Proof These rules consider the weight vector to be binary and asymmetric for the items in A.

Pascual et al. [35] already make the connection between the Kemeny rule and their utilitarian aggregation rule with swap distance, which is known to be equivalent to *Med*.

As for the egalitarian aggregation function with swap distance, this rule gives a score for each agent $i \in \mathcal{N}$ and possible outcome, which is one negative point for each item that an agent wanted and is not accepted. By a slight abuse of notation, we can define for agent $i \in \mathcal{N}$ this score as $\sum_{a_j \in A} -B_i(a_j) \times (1 - X(a_j))$, where $B_i(a_j) \in \{0, 1\}$ represents if i accepts issue a_j or not, and $X(a_j)$ represents if a_j is accepted or not in possible outcome X.

The rule returns those outcomes such that the minimum value Z of this score for any agent (i.e., the value such that all agents have at least this score) is maximal. All agents approve the same number of items in collective scheduling, i.e., $k = \sum_{a \in A} B_i(a)$ for every $i \in \mathcal{N}$. The value of Z can be at worst the number of approvals in any ballot and at best 0. Since every voter approves exactly k items of the agenda, we can add k to every agent's summation, $k + \sum_{a_j \in A} -B_i(a_j) \times (1 - X(a_j)) \geq Z + k = Z'$, for $Z' \in [0, k]$. By rearranging this inequality, we see that this is equivalent to $\sum_{a_j \in A} X(a_j) \times B_i(a_j)$ for $i \in \mathcal{N}$, which is equivalent to maximising the least number of agreements between each agent's ballot and the outcome, which is exactly what *Egal* does. □

Pascual et al. [35] also study, among others, a *tardy measure* of satisfaction, which paired with their utilitarian or egalitarian rules resemble *WMed* and *WEgal*, respectively. However, they are not instances of our judgment aggregation rules, as they rely on information specific to each agent: namely, a *due date* which may be different for each agent (thus requiring an individual weight function). Although these kinds of rules cannot be modelled in weighted judgment aggregation, and since each item of a scheduling agenda refers to *a pair* of jobs (to their preferred ordering), one way to incorporate non-binary weights would be to assign the duration of the second job as the weight of the item: i.e., the weight of $w(a_{x\prec y}) = t_y$ and $w(a_{\overline{x\prec y}}) = 0$.[9]

Moreover, if agents are allowed to submit partial schedules, we could integrate this into our model by letting them disapprove both the four elements of a pair of projects $a_{x\prec y}, a_{y\prec x}, a_{\overline{x\prec y}}, a_{\overline{y\prec x}}$. This could allow for the modelling of being indifferent between the ordering of p_x and p_y.

3.5 Collective placement

Not only does our general framework allow us to give a unifying treatment to many known problems, but it can also be used to tackle novel applications. Due to the applications being

[9] Note that this would also require adding extra items to the agenda representing the starting job in the ordering, e.g., $a_{0\prec x}, \overline{a_{0\prec x}}$ for each project $p_x \in P$, in order to take into account the duration of this starting job as well.

novel, they have not been studied independently and thus do not currently have a specific framework. Consider a situation in which agents collectively decide both which projects to accept and also how a given resource will fund them: e.g., selecting events and the rooms where they can take place, pieces of furniture to purchase and where to place them in a room, which tasks to accomplish and by which worker, and so on. The items in a *placement agenda* thus represent whether a project is funded and by which part of the resource.

Formally, let $P = \{p_1, \ldots, p_m\}$ be a set of projects (e.g., events, tasks), where $p_i \in P$ requires $c_i \in \mathbb{N}^+$ parts of a given resource (e.g., rooms, workers) to be implemented. Given a resource divisible into ℓ separate parts, a placement agenda contains a total of $\ell \cdot \sum_{p_i \in P} c_i$ elements, where each item $a^k_{i,j} \in A^+$ can be read as funding project p_i's j^{th} part with the k^{th} part of the resource. Thus, for each $p_i \in P$, for all $k \in [\ell]$ and each $j \in [c_i]$ we have $a^k_{i,j} \in A^+$, where $w_{\overline{a}} = 0$ for all $\overline{a} \in A^-$.

Feasibility constraints impose that a resource part is only used once in an outcome X (5), that a project part is only funded once (4), and either every part of the project is funded or none of it is (6).

Formally, for each $p_i \in P$ and $j \in [c_i]$ we introduce:

$$\sum_{k=1}^{\ell} X(a^k_{i,j}) \leq 1 \tag{4}$$

For each $k \in [\ell]$ we have:

$$\sum_{p_i \in P} \sum_{j=1}^{c_i} X(a^k_{i,j}) \leq 1 \tag{5}$$

Furthermore, we introduce binary variables p_i that evaluate to one when project p_i has been accepted and zero otherwise. Therefore, for each $p_i \in P$:

$$\sum_{j=1}^{c_i} \sum_{k=1}^{\ell} X(a^k_{i,j}) = c_i \cdot p_i \tag{6}$$

These constraints define a basic setup where projects may not be funded by consecutive parts of the resource.

Example 3 A company is refurbishing the floor of one of its buildings. Ten rooms will be built with the following constraints: 6 rooms for office space, at most two toilets, at most one common room and at most 5 meeting rooms (but possibly none). The employees are asked to vote on possible refurbishment plans. The resource is thus composed of 10 discrete blocks (the rooms), and the corresponding placement agenda items are $a^k_{office,j}$ for $j \in [6]$, $a^k_{common,j}$ for $j = 1$, $a^k_{toilets,j}$ for $j \in [2]$, $a^k_{meeting,j}$ for $j \in [5]$, with $k \in [10]$. An employee whose favourite floor plan is to put the six offices together first, then a toilet, and then 3 meeting rooms will thus vote $a^k_{office,k} = 1$ for $k \in [6]$, $a^7_{toilets,1} = 1$, $a^{7+j}_{meeting,j} = 1$ for $j \in [3]$, and also vote 0 on all other issues (and 1 on their negations), signalling also that they are not interested in having a common room on the floor. Of course, a nice user interface could help to visualise a vote and translate it into a ballot for this placement agenda.

4 Identifying computational complexity gaps

In the previous section, we saw that our general judgment aggregation framework is expressive enough to capture a variety of different settings. Therefore, it is a good candidate as a declarative language for collective combinatorial optimisation problems because it describes the problems in logical terms, and their algorithmic resolution is taken care of by a general solver. Moreover, we showed that many domain-specific CCO rules across the literature can be simulated by known judgment aggregation rules—either directly or with a slight generalisation to weighted and/or asymmetric agendas.

Shifting to a general framework allows for the use of additional constraints on the output to study variants of classical problems without modelling a new framework. As an example, Rey et al. [36] showed that by modelling participatory budgeting in judgment aggregation they can consider multiple resources, project dependencies or quotas on project types. Yet, this shift may come with a computational complexity gap. When there is such a gap, large instances may not become easily solvable when expressed directly in the general language using a general solver (yet, they could still be solved approximately; and small instances can be solved easily in spite of the gap). On the other hand, if there is no such gap, then generality comes for free.

This section is devoted to comparing the complexity of outcome determination for judgment aggregation rules and domain-specific CCO rules. Endriss et al. [16] give a complexity overview for standard (binary, symmetric) rules. We complement their results by classifying the complexity of the weighted counterparts of these rules and identifying the presence or absence of complexity gaps to outcome determination for multi-winner elections and participatory budgeting. We do not consider collective scheduling and collective network design here, as few complexity results are known.

We focus our study of outcome determination on the credulous (i.e., existentially quantified) version of the decision problem—the results for the skeptical (i.e., universally quantified) variants are similar, replacing classes by their co-class.

F-Credulous-Outcome-Determination (F-Cred)	
Given:	An agenda A with associated weight vector $\boldsymbol{w}$, a set of rationality and feasibility constraints modelling $\mathcal{B}_R$ and $\mathcal{B}_F$, a profile $\boldsymbol{B} \in {\mathcal{B}_R}^n$, and a distinct issue from the agenda $a^* \in A$.
Question:	Is there a feasible outcome $B \in F(\boldsymbol{B}, \Gamma_F, \boldsymbol{w})$, such that $B(a^*) = 1$?

We are now set to discuss the complexity of computing the outcome of the rules we introduced in Section 2.2, both in the general case and in two restricted cases of participatory budgeting and multi-winner elections. Table 3 summarises our results. We do not consider the complexity in the weighted setting for multi-winner elections since the weights for candidates are binary (by definition of the setting). Our findings show that for *WMed*, *WEgal* and *CC* (and we conjecture also for *Egal*), there is no increase in complexity when moving from specific CCO problems to our general formulation. This is not true for *Med* and *Rank*, whose application to participatory budgeting or multi-winner elections can be run in polynomial time. Interestingly, the restriction of the median rule *Med* to constraints whose consistency

Springer

Table 3 In this table we compare each of the rules used throughout the paper with respect to different settings: namely, judgment aggregation, multi-winner elections, and participatory budgeting

Rule	Judgment Aggregation with arbitrary weights in $\mathbb{N}_0$	Multi-winner Elections	Participatory Budgeting
Med	Θ_2^{p}-c$^{\blacklozenge}$	P	P$^{\spadesuit,\blacktriangle}$
WMed	Δ_2^{p}-c$^{\blacktriangle\oplus}$	-	Δ_2^{P}-c$^{\blacktriangle}$
Egal	Θ_2^{p}-c$^{\blacklozenge}$	coNP-h$^{\ominus}$	coNP-h$^{\ominus}$
		$\in \Theta_2^{\mathrm{P}\ominus}$	$\in \Theta_2^{\mathrm{P}\ominus}$
WEgal	Δ_2^{P}-c$^{\blacktriangle\oplus\ominus}$	-	Δ_2^{P}-c$^{\blacktriangle\oplus\ominus}$
(W)*Rank*	Δ_2^{P}-c$^{\clubsuit}$	P	P
CC	Θ_2^{P}-c$^{\heartsuit}$	Θ_2^{P}-c$^{\heartsuit}$	Θ_2^{P}-c$^{\heartsuit}$

For each rule and setting, we give the computational complexity of the (credulous) outcome determination decision problem. $^{\clubsuit}$[15]; $^{\blacklozenge}$[16]; $^{\spadesuit}$[41]; $^{\heartsuit}$[39]; $^{\blacktriangle}$[3]—hardness proof for *WMed* in PB holds for one voter, the extension of this result to other settings can be seen in $^{\oplus}$Proposition 5 and $^{\ominus}$Proposition 6; unmarked results are obviously polynomial-time computable

can be checked in polynomial-time is hard,[10] but its application to participatory budgeting and multi-winner elections is easy.

Proposition 5 *For the weighted asymmetric median rule (considering $\boldsymbol{w}_{we}^{asym}$), WMed-*Cred *is Δ_2^{p}-complete. For the binary restriction ($\boldsymbol{w}_{bin}^{sym}$ or $\boldsymbol{w}_{bin}^{asym}$), Med-*Cred *is Θ_2^{p}-complete. For participatory budgeting constraints, WMed-*Cred *is Δ_2^{p}-complete for weighted agendas ($\boldsymbol{w}_{we}^{asym}$) and in* p *for binary agendas ($\boldsymbol{w}_{bin}^{asym}$); it is in* p *for multi-winner elections.*

Proof For the upper Δ_2^{p} and Θ_2^{p} bounds, the proofs are routine. The median rule outputs judgments maximising a given value. We first identify the optimal score k^* using binary search, which needs a polynomial (resp. logarithmic) number of NP-oracle calls for weighted (resp. binary) agendas. In a final query, we may ask whether this maximal value is reached for some feasible outcome $X \in \mathcal{B}_F$ where $X(a^*) = 1$.

For participatory budgeting (and its unit-weight variant multi-winner elections), Talmon and Faliszewski [41] show that computing an outcome with the median rule (see Proposition 2) can be done in polynomial time; the result in an irresolute variant for Cred follows [3].

For the lower bounds, for the (weighted, asymmetric) participatory budgeting agenda, Baumeister et al. [3] showed that *WMed*-Cred for the weighted median rule is Δ_2^{p}-hard, even for settings with a single voter. We described in Section 3.2 how participatory budgeting can be encoded into our model. Hence, we can use the same reduction as Baumeister et al. [3], yielding Δ_2^{p}-hardness. Finally, *Med*-Cred for the binary median rule is known to be Θ_2^{p}-hard even for symmetric agendas (see [16]). The hardness result extends to asymmetric agendas since we can simulate a symmetric agenda with an asymmetric agenda (for each issue in the negative agenda we add a corresponding issue to the positive agenda, whose values must be kept consistent with the rationality and feasibility constraints). □

Proposition 6 *For the weighted asymmetric agenda ($\boldsymbol{w}_{we}^{asym}$), WEgal-*Cred *is Δ_2^{p}-complete. For the binary restriction ($\boldsymbol{w}_{bin}^{asym}$ or $\boldsymbol{w}_{bin}^{sym}$), Egal-*Cred *is Θ_2^{p}-complete.*

[10] de Haan [25] showed that hardness already holds for Horn formula constraints for binary symmetric agendas.

*For participatory budgeting, WEgal-*CRED *is* Δ_2^{p}*-complete for weighted agendas (*$\boldsymbol{w}_{we}^{asym}$*); for binary agendas (*$\boldsymbol{w}_{bin}^{asym}$*) Egal-*CRED *is in* Θ_2^{p} *and* coNP*-hard for participatory budgeting and multi-winner election constraints.*

Proof Some of the proofs for membership and hardness are similar to the proof of Proposition 5 for the (weighted) median rule. For the upper bound, we can also optimise a value which is bounded upwards to decide credulous outcome determination. Analogously, we need a polynomial number of NP-queries for weighted agendas and a logarithmic number of NP-queries for binary agendas. For the lower bounds, we begin with weighted agendas. Note that when the profile consists of a single voter $\boldsymbol{B} = (B_1)$, then for any constraint Γ_F and any weight vector $\boldsymbol{w}$, it holds that $WMed(\boldsymbol{B}, \Gamma_F, \boldsymbol{w}) = WEgal(\boldsymbol{B}, \Gamma_F, \boldsymbol{w})$. Hence, for single-voter profiles, we can reduce *WMed*-CRED for the weighted median rule to *WEgal*-CRED for the egalitarian rule. In the proof of Proposition 5, the reduction for the weighted median rule only uses a single voter; thus, we can use the same reduction, resulting in Δ_2^{p}-hardness. The result holds in particular for participatory budgeting constraints.

For the binary restriction, *Egal*-CRED is Θ_2^{p}-complete (see [16]), which for symmetric agendas still holds even if no feasibility constraint is given (see [25]). Hardness transfers to asymmetric agendas, as we can simulate symmetric agendas with asymmetric agendas. For multi-winner elections and participatory budgeting, Θ_2^{p} membership follows the same structure, while coNP-hardness comes from a straightforward reduction of the complement of the NP-complete problem EXACT COVER BY 3- SETS (see [21]). We assume that we are deciding whether a finite set of elements cannot be covered exactly (i.e., each element once) by a distinct selection of k 3-element subsets. We can reduce each element to a voter and each 3-element subset to a candidate, approved by the voters representing the contained elements. If we add another candidate, not approved by any voter, this candidate is in a winning committee of size k if and only if there is no exact cover. □

We also classify the weighted version of the ranked agenda rule: the results follow mainly from the literature.

Proposition 7 *For the rank rule (with immediate tie-breaking), WRank-*CRED *is* Δ_2^{p}*-complete, even for symmetric weights,* $\boldsymbol{w}_{bin}^{sym}$ *or* $\boldsymbol{w}_{we}^{sym}$*. For participatory budgeting and multi-winner elections, which are modelled with* $\boldsymbol{w}_{we}^{asym}$ *and* $\boldsymbol{w}_{bin}^{asym}$ *weights, respectively, WRank-*CRED *is in* P.

Proof The upper bound for *WRank*-CRED for the (weighted, asymmetric) ranked agenda rule can be derived by executing the rule, and then verifying if a given agenda item is in the final outcome. This can be done by ordering the agenda items by descending weighted support (using a fixed tie-breaking) and querying an NP-oracle in (at most) each of the m iterations (one for each item). For *WRank*-CRED the answer is yes, if the distinct agenda item is also in the outcome. The bounds are inherited from the binary, symmetric version, whose decision problem *Rank*-CRED is Δ_2^{p}-complete (see [15]).

For the (weighted) *WRank* rule there is a complexity gap between judgment aggregation and CCO problems, where we can find efficiently whether a subset of items can be extended to a feasible outcome (e.g., participatory budgeting). For a linear tie-breaking, we solve its decision problem by executing the rule and checking whether some item is in the outcome (which can be done in polynomial time if we can check the constraint efficiently). Note that $Rank(\boldsymbol{B}, \Gamma_F, \boldsymbol{w}) \in Med(\boldsymbol{B}, \Gamma_F, \boldsymbol{w})$ holds for multi-winner elections. □

Finally, Sonar et al. [39] showed Θ_2^{p}-completeness for *CC* with approval ballots in multi-winner elections. For asymmetric agendas, the lower bound inherits to judgment aggregation, while the upper bound can be shown analogously to the upper bound of *Med*-CRED in the proof of Proposition 5.

Proposition 8 *CC-CRED is Θ_2^{p}-complete with either $\boldsymbol{w}_{bin}^{asym}$ or $\boldsymbol{w}_{we}^{asym}$ weights.*

In classical judgment aggregation, where the agenda is symmetric and every rational judgment is feasible, the *CC* rule becomes degenerate, because the only time an agent is not happy with an outcome is when it is the complement of their ballot. Although the usability of the Chamberlin-Courant rule is limited for symmetric agendas, we still provide tight bounds for CC-CRED on symmetric weights.

In the following proofs, we denote by $\overline{X}$ the complement of a vector X, i.e., a vector where all 0s and 1s have been swapped and by $CC_{score}(\boldsymbol{B})$ the number of voters who are simultaneously satisfied by a given outcome $X \in CC(\boldsymbol{B})$. Also, for any vector X we denote $Occur(X, \boldsymbol{B})$ the number of occurrences of X in $\boldsymbol{B}$, *i.e.*, the number of voters i such that $B_i = X$. We start with the following observation.

Observation 9 *For an arbitrary agenda A with symmetric weights $\boldsymbol{w}_{we}^{sym}$, and assuming $\Gamma_R = \Gamma_F$, then*

1. *$CC_{score}(\boldsymbol{B}) = n - \min_{X \in \mathcal{B}_R} Occur(\overline{X}, \boldsymbol{B})$.*
2. *If $CC_{score}(\boldsymbol{B}) = n$ then $X \in CC(\boldsymbol{B})$ if and only if $X \in \mathcal{B}_R$ and $Occur(\overline{X}, \boldsymbol{B}) = 0$, that is, for each $i \in \mathcal{N}$, $B_i \neq \overline{X}$.*
3. *If $CC_{score}(\boldsymbol{B}) < n$, then for all $X \in CC(\boldsymbol{B})$ there is some $i \in \mathcal{N}$ with $B_i = \overline{X}$.*

Proof For point 1, note that we have $CC(\boldsymbol{B}) = \mathrm{argmin}_{X \in \mathcal{B}_R} Occur(\overline{X}, \boldsymbol{B})$ by definition. Therefore, $CC_{score}(\boldsymbol{B}) = n - \min_{X \in \mathcal{B}_R} Occur(\overline{X}, \boldsymbol{B})$.

For point 2, assume $CC_{score}(\boldsymbol{B}) = n$. Then, point 1 is $\min_{X \in \mathcal{B}_R} Occur(\overline{X}, \boldsymbol{B}) = 0$, i.e., there exists a feasible vector X such that for all i, $B_i \neq \overline{X}$.

For point 3, assume $CC_{score}(\boldsymbol{B}) < n$. Then, by point 1, for each $X \in \mathcal{B}_R$, we have $Occur(\overline{X}, \boldsymbol{B}) > 0$, which means that there is an $i \in \mathcal{N}$ such that $B_i = \overline{X}$. □

We can now prove that credulous outcome determination becomes coDP-complete, where coDP $= \{L \cup L' \mid L \in$ NP, $L' \in$ coNP$\}$ (see [34]). Less formally, DP (resp. coDP) is the class of decision problems that can be written as the intersection (resp. the union) of a problem in NP and a problem in coNP. A canonical DP-complete problem is SAT- UNSAT by Papadimitriou and Yannakakis [34]: an instance (ϕ, ψ) of the problem consists of two boolean formulas ϕ and ψ, and the question is whether ϕ is satisfiable while ψ is unsatisfiable.

Proposition 10 *If $\Gamma_R = \Gamma_F$ and the agenda weights are restricted by $\boldsymbol{w}_{bin}^{sym}$ or $\boldsymbol{w}_{we}^{sym}$, CC-CRED is coDP-complete.*

Proof We begin with the upper bound. Observation 9 gives us an algorithm for computing $CC(\boldsymbol{B})$: if there is $X \in \mathcal{B}_R$ such that for all i, $B_i \neq \overline{X}$, then output all such vectors X; else output $\mathrm{argmin}_{i \in \mathcal{N}} Occur(\overline{B_i}, \boldsymbol{B})$. Therefore, there exists a feasible outcome $X \in CC(\boldsymbol{B})$ such that $X(a^*) = 1$ if (at least) one of these two conditions is met:

1. There is $X \in \mathcal{B}_R$ such that for all i, $B_i \neq \overline{X}$, and $X(a^*) = 1$.
2. There is no $X \in \mathcal{B}_R$ such that for all i, $B_i \neq \overline{X}$, and there is an i with $B_i(a^*) = 1$ such that $Occur(\overline{B_i}, \boldsymbol{B}) \leq Occur(\overline{B_j}, \boldsymbol{B})$ for all j.

The set of all instances meeting condition 1 (resp. 2) is a problem in NP (resp. coNP), therefore CC-CRED restricted to agenda weights in $\boldsymbol{w}_{bin}^{sym}$ or $\boldsymbol{w}_{we}^{sym}$ is in coDP.

For the lower bound, we reduce the DP-complete problem SAT- UNSAT to the complement of CC-CRED, mapping YES-instances of SAT- UNSAT to NO-instances of CC-CRED and vice-versa. Given any SAT- UNSAT instance (ϕ, ψ), we construct an instance for the complement of CC-CRED as follows.

Let X (respectively, Y) contain all variables appearing in ϕ (respectively, ψ) and their negations. We assume without loss of generality that X and Y are disjoint, i.e., $X \cap Y = \emptyset$ and set the agenda to be $A = X \cup Y \cup \{a^*, \overline{a^*}, b, \overline{b}\}$, where a^* and b are newly introduced variables; thus, $m = |X| + |Y| + 4$. The rationality and feasibility constraint is defined as $\Gamma_R = \Gamma_F = \alpha \vee \beta \vee \gamma \vee \delta$, where $\alpha = \overline{a^*} \wedge \phi \wedge b$, $\beta = a^* \wedge \psi \wedge \overline{b}$, $\gamma = \bigwedge_{a \in A^+} a$ and $\delta = \bigwedge_{\overline{a} \in A^-} \overline{a}$.[11] Note that a vote on all positive literals satisfies only γ, while its complement only satisfies δ. The profile $\boldsymbol{B}$ consist of three voters, for simplicity we give their votes on A^+: $\boldsymbol{B}^+ = (\{1\}^{m/2}, \{1\}^{m/2}, \{0\}^{m/2})$, whereby $\{1\}^{m/2}$ we denote for simplicity an agent approving all the items in A^+ and by $\{0\}^{m/2}$ an agent approving all the items in A^-. We claim that a^* is not in any outcome of $CC(\boldsymbol{B})$ if and only if (ϕ, ψ) is a YES-instance.

Without considering further satisfying assignments that do not appear in the profile $\boldsymbol{B}$, issue a^* will be a temporary winner (i.e., it appears in the judgment whose complement occurs in the profile the least number of times). There is only one way that prevents a^* from being in any outcome: i.e., if there exists a judgment $X \in \mathcal{B}_F$ with $X(a^*) = 0$ and $X \neq \{0\}^m$, while there exists no judgment $X \in \mathcal{B}_F$ with $X(a^*) = 1$ and $X \neq \{1\}^m$. By construction, this can only hold if and only if α is satisfiable and β is unsatisfiable. It is easy to see that α is satisfiable if and only if ϕ is satisfiable, while β is unsatisfiable if and only if ψ is unsatisfiable. □

5 Conclusion

We have four main take-home messages: *(i)* when looking for a declarative language to express various CCO problems, judgment aggregation is a good candidate; *(ii)* however, we need a slight generalisation where issues are weighted, and weights may be asymmetric—this generalisation allows for specific CCO problems to be seen through the lens of judgment aggregation; *(iii)* several rules studied for specific CCO problems, namely participatory budgeting, multi-winner elections, collective scheduling, and collective network design, are instances of the general settings—this shows strong connections between two specific 'sister' rules, that are instances of the same general rules and share common normative properties; *(iv)* in about half of the cases considered, the generalisation does not come with a complexity increase.

In the Appendix we present an experimental case study intended to show the applicability of our proposed general judgment aggregation framework. We report on experiments using an ILP solver to compute the result of collective networking problems, comparing the running time of three of the proposed rules in different graph configurations and vote generation models.

[11] Observe that we can adapt the construction of the constraints to be linear inequalities rather than propositional formulas. Assume ϕ and ψ are in conjunctive normal form (CNF), for which SAT- UNSAT is still DP-hard. Then, formulas α, β, γ and δ are also in CNF. For any two formulas $\varphi_1 = \bigwedge_{i=1}^{n} c_1^i$ and $\varphi_2 = \bigwedge_{j=1}^{m} c_2^j$ in CNF, we can transform $\varphi_1 \vee \varphi_2$ into CNF in polynomial time. By repeatedly using laws of distributivity, the resulting CNF formula contains a clause for every pair of clauses (one from each of the two formulas), i.e., $\varphi_1 \vee \varphi_2 \equiv \bigwedge_{i=1}^{n} \bigwedge_{j=1}^{m} (c_1^i \vee c_2^j)$. Hence, we can transform Γ_F into CNF using space that is polynomial in the size of Γ_F. Finally, we can express any formula in CNF as ILP constraints by adding an inequality for each of its clauses (the sum of its variables' values must be at least one).

Appendix: General solvers

Our general framework of weighted asymmetric judgment aggregation can not only be used for theoretical comparisons of rules for specific CCO settings but also to obtain a modular implementation of the rules: by simply plugging in the constraints, one can focus on a particular application. Although there is no specific solver for *weighted asymmetric* judgment aggregation, integer linear programming (ILP) is an ideal choice for such a solver. Each rule given in this paper (see Section 2.2) can be translated into an ILP formulation, as shown in the following subsection.

To the best of our knowledge, this is the first ILP formulation for judgment aggregation rules—prior to this, the only other implementations have used answer set programming [26] and SAT solvers [9]. One of the benefits of using the ILP formalism is the ability to rely on its vast literature and efficient solvers. Furthermore, we conjecture that many more judgment aggregation rules (and their weighted extensions) can be expressed as an ILP, which may not be straightforward when using other solvers that do not use cardinal weights (such as SAT solvers, where weights might have to be simulated). A benefit of ILP solvers is that many are efficient, and constraints are expressed compactly as sets of inequalities. Moreover, the use of JA as a general model for CCO problems can be motivated by the natural translation of the rules into ILP. This is unlike some of the other general solvers where the translations of the rules are far more involved.

Note that the constraints for the CCO problems described in Section 3 are already presented as sets of linear inequations, allowing us to study those in ILP directly.

Integer Linear Program Formulations for Weighted Asymmetric Judgment Aggregation Rules

We give an ILP formulation for each of our studied rules. In the following, each agenda item $a_j \in A$ is given as binary variable $a_j \in \{0, 1\}$ and we assume that we are given a (possibly empty) feasibility constraint Γ_F as a set of linear inequalities.

The (weighted) median rule

Following Definition 1, we can formulate *WMed* as the following ILP.

$$\begin{array}{ll} \text{Maximise} & \sum_{a_j \in A} \sum_{i=1}^{n} w_{a_j} \times B_i(a_j) \times a_j \\ \text{Subject to} & \Gamma_F \\ & \forall a_j \in A : a_j \in \{0, 1\} \\ & \forall a_j, \overline{a_j} \in A : a_j = 1 - \overline{a_j} \end{array} \tag{A1}$$

The (weighted) egalitarian rule

To express the egalitarian rule in ILP we use an additional variable Z, which represents the lowest score of any agent (and thus should be maximised). This maximisation is done subject to the feasibility constraints (Γ_F) and the intersection of the outcome assignment over the agenda A with respect to each agent's ballot, which must be greater than or equal to Z. A similar ILP formulation in the context of participatory budgeting was given by Sreedurga et al. [40].

$$
\begin{array}{lc}
\text{Maximise} & Z \\
\text{Subject to} & \Gamma_F \\
& \text{for } i \in \mathcal{N} : \sum_{a_j \in A} w_{a_j} \times B_i(a_j) \times a_j \geq Z \\
& \forall a_j \in A : a_j \in \{0, 1\} \\
& Z \in [0, \sum_{a \in A^+} w_a] \\
& \forall a_j, \overline{a_j} \in A : a_j = 1 - \overline{a_j}
\end{array} \tag{A2}
$$

The (weighted) ranked agenda rule

Note that *Rank* and *WRank* can be computed efficiently for easy-to-solve constraints (e.g., a participatory budgeting constraint). Nevertheless, we provide an ILP formulation to compute the outcome in general. In a pre-processing step, we compute the order in which decisions over the agenda are made, i.e., descending by the agents (weighted) support, where ties are broken alphabetically. For $a \in A$, let π_a be the number of items that are ranked after a in that order. Then, for the ILP formulation of *WRank* it is sufficient to ensure that it is always preferred to include issues that are processed earlier over all issues that are processed later.

$$
\begin{array}{lc}
\text{Maximise} & \sum_{a_j \in A} \sum_{i=1}^{n} 2^{\pi_{a_j}} \times a_j \\
\text{Subject to} & \Gamma_F \\
& \forall a_j \in A : a_j \in \{0, 1\} \\
& \forall a_j, \overline{a_j} \in A : a_j = 1 - \overline{a_j}
\end{array} \tag{A3}
$$

The Chamberlin-Courant rule

For an ILP formulation of the Chamberlin-Courant rule, we take inspiration from Talmon and Faliszewski [41]—see also [38] for a translation of the multi-winner election variant into ILP. We need an extra variable c_i for each agent $i \in \mathcal{N}$ that will model their satisfaction. In particular, c_i will be set to 1 if there is a project in the outcome which the agent approves.

$$
\begin{array}{lcc}
\text{Maximise} & & \sum_{i=1}^{n} c_i \\
\text{Subject to} & & \Gamma_F \\
& \forall i \in \mathcal{N} : & \sum_{a_j \in A} B_i(a_j) \times a_j \times w_{a_j} \geq c_i \\
& \forall i \in \mathcal{N} : & c_i \in \{0, 1\} \\
& \forall a_j \in A : & a_j \in \{0, 1\} \\
& \forall a_j, \overline{a_j} \in A : a_j = 1 - \overline{a_j} &
\end{array} \tag{A4}
$$

A Case Study for General Solvers: Collective Networking

In this section, we provide a case study for the implementation of the collective networking problem (described in Section 3.3) without a budget constraint, comparing the processing time of three (binary) rules: the median rule (*Med*), the egalitarian rule (*Egal*), and the Chamberlin-Courant rule (*CC*). We do not study the (weighted) ranked agenda rules, as they are solvable in polynomial time (if it can be checked in polynomial time whether a partial assignment can be extended to a full assignment satisfying the given constraint). The implementation used the open-source *GNU Octave* software [12], and its standard ILP solver `glpk`, using two-phase primal simplex method. Our implementation is modular: i.e., the same set-up can be altered to account for any CCO problem by changing the problem-specific components (the agenda, the constraints, and the CCO rule).

Recall that in the collective networking problem, we have a graph $G = (V, E)$ and the agents vote on the edges E in the corresponding agenda, while the rule has to find a collective spanning tree. Regarding the constraints, we allow for any possible ballot and we only impose that the outcome must be a spanning tree, thus Γ_F is composed of the ILP constraints expressed in (1), (2) and (3).

We generate the underlying network in the form of 49 connected graphs $G = (V, E)$ with number of nodes varying between 6 and 8, i.e., $V \in [6, 8]$. For each value of V we generate connected graphs with $E \in \left[|V| - 1, \frac{|V|(|V|-1)}{2}\right]$: i.e., the graphs vary from being trees (for $|E| = |V| - 1$) to being complete (for $|E| = \frac{|V|(|V|-1)}{2}$). Each graph is randomly generated as follows, for a given $|V|$ and $|E| = e$.

We initially let S and S' be two sets such that one element v_0 of V is in S (i.e., $v_0 \in S$) and $S' = V \backslash \{v_0\}$. The set S and S' can be seen as the set of connected and unconnected nodes, respectively. The algorithm iteratively chooses, at random, an item $(v_i, v_j) \in S \times S'$, moves v_j from S' to S, and adds (v_i, v_j) to the set of edges E. When S' is empty, we randomly add $e - (|V| - 1)$ extra edges from $(V \times V) \setminus E$.

We let $|\mathcal{N}| = 100$. On each generated graph G we create 10 *base profiles*. Each base profile (bp), is an $n \times E$ matrix, where for each $i \in \mathcal{N}$ we have that $bp_i \in (0, 1]^E$: that is, for each item of the agenda a real number between 0 and 1 is assigned to represent the acceptance rate of an issue by an agent. Each base profile is then transformed into 9 new profiles as follows, under the variant of the p-impartial culture model presented by [6]. According to this model, when generating approval voting ballots, one can assume that every agent independently approves each item of the agenda with probability p. Therefore, for each base profile (generated for each graph) we create 9 profiles, one for every $p \in \{0.1, \ldots, 0.9\}$. If the value of an entry b in the base profile is such that $b \leq p$, then in the created profile the agent's preference on this issue will be an approval, and an abstention otherwise.

Therefore, for each of our generated graphs (a total of 49 graphs), we created a total of 90 profiles originating from 10 base profiles and the 9 probabilities from our p-impartial culture assumption. Thus, we ran the three CCO rules on 4410 instances. We then run the ILP solver on each profile and its related graph to find an outcome for the three rules: *Med*, *Egal* and *CC*.

For each generated problem instance we measured the processing times of computing the outcome of three rules, given varying levels of the p-impartial culture and number of nodes in the underlying initial network.[12]

We begin by comparing the run-times of the three CCO rules with respect to four levels of acceptance, $p \in \{0.2, 0.4, 0.6, 0.8\}$ and varying the number of nodes in the graph (we only highlight four of our nine values of p). Figure 2 shows, for each value of p, the mean processing times over all profiles and all generated networks varying the number of nodes of the network $V \in \{6, 7, 8\}$. In Fig. 2 we use a $\log_2$-scale on the y-axis due to the exponential increase in processing times.

If we focus on *CC*, we observe that the run-time is inversely proportional to the acceptance level p, confirming the intuition that finding a collective spanning tree with *CC* is more difficult with sparse ballots.

The *Med* rule is slower to compute than the other rules for almost all p values (with some exceptions for small values of p against *CC*). Observe that without additional constraints

[12] The experiments ran on an Intel i7 processor at 4.2 GHz with 4 physical and 8 logical cores and 32 GB of memory. At any time, six instances were computed in parallel.

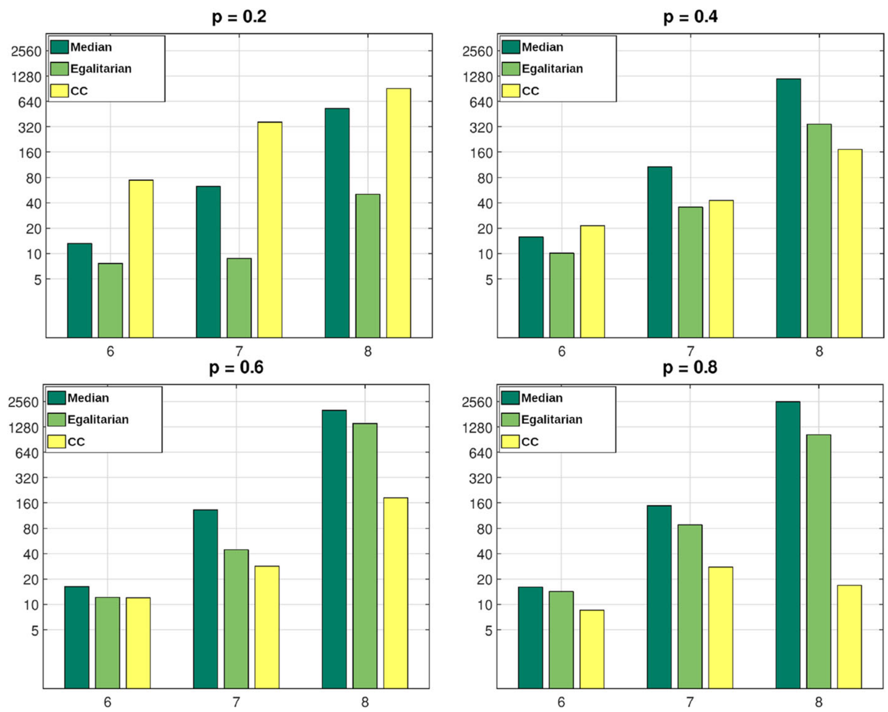


Fig. 2 Mean processing time for the *Med*, *Egal* and *CC* rules applied on the spanning tree problem. The x axis represents the number of nodes in the graph (from 6 to 8); the y axis represents the mean processing time (milliseconds) on a logarithmic scale ($\log_2$-scale). Each figure shows the mean results for a specific level of acceptance $p \in \{0.2, 0.4, 0.6, 0.8\}$

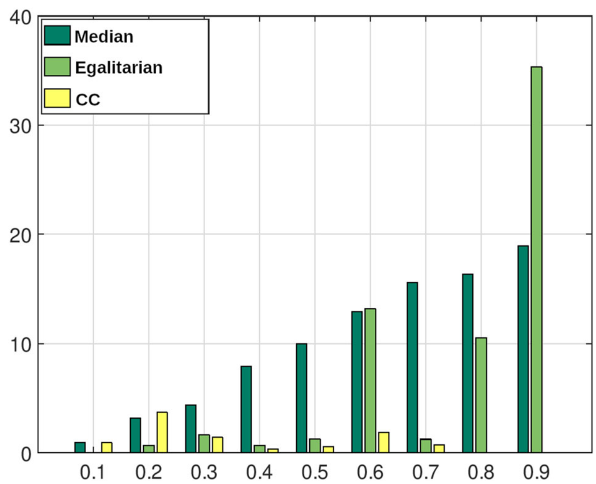


Fig. 3 Mean processing time (seconds) for the *Med*, *Egal* and *CC* rules applied on the spanning tree problem, for $p \in \{0.1, \ldots, 0.9\}$, on a complete graph with 8 nodes

on the budget, the *Med* rule is equivalent to finding the maximum spanning tree where the weights of the edges are determined by how many voters approved them. Therefore, the complexity of computing the outcome of *Med* is proportional to $N \cdot p$, i.e., the weights of the edges. A future step would be to check if the same increase in run-time can be observed for the related rule *WMed*.

Finally, the run-time of *Egal* increases steadily with the number of nodes when $p = 0.2, 0.4$, while for $p = 0.6, 0.8$ it increases at a much quicker rate. This can be explained by referring back to (A2), where we see that *Egal* maximises the value of variable Z, whose upper bound is the minimum number of items that any agents has approved. Therefore, when p is low, so is the upper bound on Z, reducing the search space; while when p is high there are more values which Z can take.

The same experiments for networks with 9 or more nodes resulted in some of the instances not completing before the chosen time-out of 1200 seconds. The observed behaviour varied with each rule: for *Med* the non-completing instances corresponded to the graphs that were close to being complete, whereas for *Med* and *Egal* the pattern was more complex, and it seemed to depend on both the structure of the graph as well as the profile of individual ballots.

Figure 3 presents the mean run-times of the three chosen rules for varying levels of acceptance p, starting from a fixed complete network with $|V| = 8$ and $|E| = 28$. Each bar represents the mean run-time for 10 different profiles, for a total of 90 instances. The figure highlights the exponential increase in running time of *Egal*, and confirms the observation that *CC* is easy on complete graphs.

Acknowledgements We thank the AMAI reviewers, as well as the reviewers of M-PREF 2022, for their valuable feedback. This work was supported by the French "Agence Nationale de la Recherche", both as part of the 'Investissements d'avenir' programme, reference ANR-19-P3IA-0001 (PRAIRIE 3IA Institute), and by the ANR JCJC project SCONE (ANR 18-CE23-0009-01). It was also supported by the NRW project "Online Participation".

Data Availability The datasets for the experiments in the appendix generated and analysed during the current study are available from the corresponding author on reasonable request.

Code Availability The code for performing the experiments in the appendix is available from the corresponding author on reasonable request.

References

1. Abdelmaguid, T.F.: An efficient mixed integer linear programming model for the minimum spanning tree problem. Mathematics **6**(10), 183 (2018)
2. Aziz, H., Faliszewski, P., Grofman, B. et al.: Egalitarian committee scoring rules. In: Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI-2018) (2018)
3. Baumeister, D., Boes, L., Hillebrand, J.: Complexity of manipulative interference in participatory budgeting. In: Proceedings of the 7th international conference on algorithmic decision theory (ADT-2021) (2021)

4. Botan, S., de Haan, R., Slavkovik, M. et al.: Egalitarian judgment aggregation. In: Proceedings of the 20th international conference on autonomous agents and multiagent systems (AAMAS-2021) (2021)
5. Brams, S.J., Kilgour, D.M., Sanver, M.R.: A minimax procedure for electing committees. Public Choice **132**(3–4), 401–420 (2007)
6. Bredereck, R., Faliszewski, P., Kaczmarczyk, A. et al.: An experimental view on committees providing justified representation. In: Proceedings of the 28th international joint conference on artificial intelligence (IJCAI-2019) (2019)
7. Chamberlin, J.R., Courant, P.N.: Representative deliberations and representative decisions: proportional representation and the Borda rule. The American Political Science Review, pp. 718–733 (1983)
8. Chingoma, J., Endriss, U., de Haan, R.: Simulating multiwinner voting rules in judgment aggregation. In: Proceedings of the 21st international conference on autonomous agents and multiagent systems (AAMAS-2022) (2022)
9. Conati, A., Niskanen, A., Järvisalo, M.: Sat-based judgment aggregation. In: Proceedings of the 2023 international conference on autonomous agents and multiagent systems (AAMAS-2023) (2023)
10. Darmann, A., Klamler, C., Pferschy, U.: Computing spanning trees in a social choice context. In: Proceedings of the 2nd international workshop on computational social choice (COMSOC-2008) (2008)
11. Darmann, A., Klamler, C., Pferschy, U.: Maximizing the minimum voter satisfaction on spanning trees. Math. Soc. Sci. **58**(2), 238–250 (2009)
12. Eaton, J.W., Bateman, D., Hauberg, S. et al.: GNU Octave version 5.2.0 manual: A high-level interactive language for numerical computations (2020). https://www.gnu.org/software/octave/doc/v5.2.0/
13. Endriss, U.: Judgment aggregation. In: Brandt, F., Conitzer, V., Endriss, U., et al. (eds.) Handbook of Computational Social Choice. chap 17 (2016)
14. Endriss, U.: Judgment aggregation with rationality and feasibility constraints. In: Proceedings of the 17th international conference on autonomous agents and MultiAgent systems (AAMAS-2018) (2018)
15. Endriss, U., de Haan, R.: Complexity of the winner determination problem in judgment aggregation: Kemeny, Slater, Tideman, Young. In: Proceedings of 14th international conference on autonomous agents and multiagent systems (AAMAS-2015) (2015)
16. Endriss, U., de Haan, R., Lang, J., et al.: The complexity landscape of outcome determination in judgment aggregation. Journal of Artificial Intelligence Research **69**, 687–731 (2020)
17. Escoffier, B., Gourvès, L., Monnot, J.: Fair solutions for some multiagent optimization problems. Auton. Agent. Multi-Agent Syst. **26**(2), 184–201 (2013)
18. Everaere, P., Konieczny, S., Marquis, P.: On egalitarian belief merging. In: Proceedings of the 14th international conference on the principles of knowledge representation and reasoning (KR-2014) (2014)
19. Everaere, P., Konieczny, S., Marquis, P.: Belief merging versus judgment aggregation. In: Proceedings of the 14th international conference on autonomous agents and MultiAgent systems (AAMAS-2015) (2015)
20. Fluschnik, T., Skowron, P., Triphaus, M., et al.: Fair knapsack. In: Proceedings of the 33rd AAAI conference on artificial intelligence (AAAI-2019) (2019)
21. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H, Freeman and Company (1979)
22. Goel, A., Krishnaswamy, A.K., Sakshuwong, S. et al.: Knapsack voting. Collective Intelligence 1 (2015)
23. Grandi, U., Endriss, U.: Lifting integrity constraints in binary aggregation. Artif. Intell. **199–200**, 45–66 (2013)
24. Grossi, D., Pigozzi, G.: Judgment Aggregation: A Primer. Synthesis Lectures on Artificial Intelligence and Machine Learning (2014)
25. de Haan, R.: Hunting for tractable languages for judgment aggregation. In: Proceedings of the 16th international conference on principles of knowledge representation and reasoning (KR-2018) (2018)
26. de Haan, R., Slavkovik, M.: Answer set programming for judgment aggregation. In: Proceedings of the 28th international joint conference on artificial intelligence (IJCAI-2019) (2019)
27. Klamler, C., Pferschy, U., Ruzika, S.: Committee selection under weight constraints. Math. Soc. Sci. **64**(1), 48–56 (2012)
28. Konieczny, S., Pino Pérez, R.: Logic based merging. J. Philos. Log. **40**(2), 239–270 (2011)
29. Lackner, M., Skowron, P.: Multi-Winner Voting with Approval Preferences. Springer Briefs in Intelligent Systems, Springer (2023)
30. Lang, J., Slavkovik, M.: Judgment aggregation rules and voting rules. In: Proceedings of the 3rd International Conference on Algorithmic Decision Theory (ADT-2013) (2013)
31. Lang, L., Pigozzi, P., Slavkovik, M., et al.: A partial taxonomy of judgment aggregation rules and their properties. Soc. Choice Welfare **48**(2), 327–356 (2017)
32. Nehring, K., Pivato, M.: The median rule in judgement aggregation. Econ. Theor. **73**(4), 1051–1100 (2022)

33. Nehring, K., Pivato, M., Puppe, C.: The Condorcet set: Majority voting over interconnected propositions. Journal of Economic Theory **151**, 268–303 (2014)
34. Papadimitriou, C.H., Yannakakis, M.: The complexity of facets (and some facets of complexity). In: Proceedings of the 14th annual ACM symposium on theory of computing (1982)
35. Pascual, F., Rzadca, K., Skowron, P.: Collective schedules: Scheduling meets computational social choice. In: Proceedings of the 17th international conference on autonomous agents and MultiAgent systems (AAMAS-2018) (2018)
36. Rey, S., Endriss, U., de Haan, R.: Designing participatory budgeting mechanisms grounded in judgment aggregation. In: Proceedings of the 17th international conference on principles of knowledge representation and reasoning (KR-2020) (2020)
37. Skowron, P., Faliszewski, P.: Chamberlin-Courant rule with approval ballots: Approximating the Max-Cover problem with bounded frequencies in FPT time. Journal of Artificial Intelligence Research **60**, 687–716 (2017)
38. Skowron, P., Faliszewski, P., Slinko, A.M.: Achieving fully proportional representation: Approximability results. Artif. Intell. **222**, 67–103 (2015)
39. Sonar, C., Dey, P., Misra, N.: On the complexity of winner verification and candidate winner for multi-winner voting rules. In: Proceedings of the 29th international joint conference on artificial intelligence (IJCAI-2020) (2020)
40. Sreedurga, G., Ratan Bhardwaj, M., Narahari, Y.: Maxmin participatory budgeting. In: Proceedings of the 31st international joint conference on artificial intelligence (IJCAI-22) (2022)
41. Talmon, N., Faliszewski, P.: A framework for approval-based budgeting methods. In: Proceedings of the 33th AAAI conference on artificial intelligence (2019)

CHAPTER 9

DISTORTION IN ATTRIBUTE APPROVAL COMMITTEE ELECTIONS

In this chapter, we explore attribute approval elections and distortion. Formally, for several voting rules relying on attribute approval ballots, we investigate the decline of the voters' satisfaction, that is possibly arising from considering derived preferences, given in the form of either approval ballots, weak ordinal rankings, or cardinal preferences.

9.1 Summary

In this article, we extend a recent study on attribute approval elections [102], to measure whether the considered, slightly more expressive ballot format for multiwinner elections can derive better decisions. For this, we consider specific use-cases, where each candidate is associated with exactly one attribute (e.g., a quality criteria) for a variety of different categories. Then, instead of approving candidates directly, a voter's attribute approval ballot specifies, which subset of attributes are desired for each category. The considered model is introduced formally in Subsection 2.2.2.

A modular setup provides a total of six aggregation rules, generalizing popular voting mechanisms for approval-based preferences to the attribute approval context. As a first step, to measure the potential upside of voting on attributes instead of candidates, we introduce derivation methods to model reasonable ways voters might have voted, when given a more common ballot format. Assuming the appeal of a candidate is connected to the satisfaction of underlying quality criteria, we consider the following derivation methods. For cardinal preferences, we assume it is reasonable for voters to assign each candidate a score linear in the number of satisfied attributes. For (weak) ordinal preferences, a (weak) ranking is induced by cardinal preferences. And for approval ballots, voters might approve those candidates, where the number of satisfied attributes surpasses a given threshold.

As a second step, we formally introduce a measure of distortion[53] for attribute approval elections and a given derivation method. That is, we study the worst-case factor, by which the voters' overall satisfaction may decline, if ballots are cast in different (derived) preference formats, but aggregated using attribute approvals.

Lastly, along with supplementing formal proofs, which were omitted due to space in our publication [21], we complement our study by two aspects in Appendix A.2. On the one hand, we present a formal encoding to model weighted attribute approval elections and briefly discuss how our results on distortion extend to this weighted generalization. On the other hand, we investigate the limits of expressiveness for attribute approval ballots. Formally, for an individual scoring function and any attribute approval election with a sufficiently large number of distinct attributes any candidate may satisfy, we study what kind of weak linear rankings over the set of k-committees can be induced by a voter's ballot.

9.2 Reflection on Initial Research Goals

In this article, we addressed three of our initial research questions, introduced in Chapter 3. Most notably, this work is exactly in the scope of Question Q5, as we studied how the more expressive ballot format of attribute approval ballots may (or may not) lead to better decisions. For different derivation methods, we showed that the distortion for an attribute approval election rule and a given derivation method may range from unbounded, over linear in the number of attribute categories or committee size, to undistorted. Overall, this is an indication for when it may be beneficial to switch to the more expressive ballot format. We addressed Question Q4 by generalizing prominent voting rules for approval preferences. As pointed out by Kagita, Pujari, Padmanabhan, Aziz, and Kumar [102], we can easily model candidate approval ballots by having only one category, which holds exactly one unique attribute for each candidate. Hence, lower bounds on the computational complexity are inherited from the candidate approval setting, implicitly addressing Question Q2 to some degree.

9.3 Publication

This work has been published and presented as an extended abstract at the *22nd International Conference on Autonomous Agents and Multiagent Systems*.

[21] D. Baumeister and L. Boes. "Distortion in Attribute Approval Committee Elections". In: *Proceedings of the 22nd International Conference on Autonomous Agents and Multiagent Systems*. IFAAMAS, 2023, pp. 2649–2651

[53] Distortion, introduced by Procaccia and Rosenschein [147], is usually studied when cardinal values capture the voters' preferences exactly, while ballots are cast in a less expressive format (see also [2, 38]).

9.4 Personal Contribution

This work has been initiated by my supervisor Dorothea Baumeister, as an academic interest she considered to pursue for a considerable time. The conception, development of our notion for distortion, and writing of this article was done jointly in equal parts by Dorothea Baumeister and myself. All technical results are my contributions under the noteworthy guidance of Dorothea Baumeister.

Distortion in Attribute Approval Committee Elections

Extended Abstract

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

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

ABSTRACT

In attribute approval elections, the task is to select sets of winning candidates, while each candidate satisfies a variety of attributes in different categories (e.g., academic degree, work experience, location). Every voter specifies, which attributes in each category are desirable for a candidate, whereas each candidate might satisfy only some of the attributes. In this paper, we study questions of distortion in attribute approval committee elections. We introduce different methods to derive approval ballots, ordinal preferences, or cardinal preferences from a given attribute approval ballot. Then for a given voting method, assuming only a derived preference is provided, we compute the ratio of the voters' satisfaction for the worst possible committee, with the satisfaction of the actual winning committee, given the attribute approval ballots.

KEYWORDS

Distortion; Multiwinner Voting; Preference Aggregation

ACM Reference Format:

Dorothea Baumeister and Linus Boes. 2023. Distortion in Attribute Approval Committee Elections: Extended Abstract. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 3 pages.

1 INTRODUCTION

Many different situations require the selection of a committee. For example a committee of people, but also a selection of movies on a plane, or the selection of dishes in a menu. See Faliszewski et al. [7] for a detailed discussion on different types of committee elections. Common ways to represent preferences in such elections are approval votes or rankings over the candidates, see Zwicker [13]. These preferences focus on single candidates, hence in committee elections the voters are not able to represent their opinion about possible outcomes, i.e. committees, of the election. Only if the number of candidates is small, it may be feasible to elicit preferences over all different committees. There are approaches to compactly represent complex preferences like CP-nets studied by Boutilier et al. [4], however they require expert knowledge. Another aspect is, that in the composition of the committee, the attributes of a candidate may be more important than the person (or object) itself. For an expert committee, the voter may want to ensure knowledge in some specific field, rather than a specific person to be present. Thus we study committee elections by focusing on the attributes of the candidates. Different vote representations and corresponding aggregation methods have received little attention so far (see [3], [11], [8] and [2]). We adopt the approach of Kagita et al. [9] and assume that there are different categories and each candidate has an attribute for each of them. The voters do not vote directly on the candidates, but approve a set of attributes for each category. The decision on single attributes may be easier for voters than to decide between a high number of candidates or committees. For winner determination we follow the approach by Kagita et al. [9], and aggregate the votes on the attributes to derive a decision on the candidates. In particular, we generalize voting rules for committee elections with candidate approvals to the attribute approval setting. We study the question, whether preferences on attributes of the candidates may provide better results than other commonly used types of preferences. We consider fixed-size committee elections where the candidates are associated with attributes from different categories and the votes are sets of approved attributes for each category. We investigate for six different rules on how these ballots may be aggregated to elect a committee. Then we define a notion of distortion, which measures how much the loss of information — from casting approval ballots, rankings, or cardinal preferences instead of attribute approval preferences — may affect the voters' satisfaction with the outcome negatively. This concept was formally introduced by Procaccia and Rosenschein [12] for underlying cardinal preferences, where distortion was measured with respect to a social choice function that aggregates derived ordinal preferences. In contrast, we consider underlying attribute approval preferences and measure the distortion between elections that have identically derived (e.g., ordinal) preferences. Closely related to our work is the study of so-called diversity constraints (see [6] and [10]), where the candidates have attributes, and the final committee has to fulfill certain requirements regarding the attributes.

2 PRELIMINARIES

For an integer i let $[i] = \{1, 2, \dots, i\}$ and for a set C let $\mathcal{P}(C)$ denote the power set of C and $\mathcal{P}_k(C) = \{W \subseteq C : |W| = k\}$ the set of all k-committees with respect to C. We follow Kagita et al. [9], who initiated a study on selecting committees using attribute approvals.

Definition 1. *Let $\mathcal{E}$ be the set of all attribute approval elections. A single such election is given by a tuple $(D, C, V) \in \mathcal{E}$, with*

- *$D = D^1 \times \dots \times D^d$, where $D^1, \dots, D^d$ for $d \in \mathbb{N}$ are attribute domains. We assume $|D^j| \geq 2$ and $D^j \cap D^h = \emptyset$ for all $j \neq h$.*
- *$C = \{c_1, \dots, c_m\}$ is a set of m candidates, where each candidate is associated with attributes from different categories, i.e., each candidate $c_i \in C$ satisfies exactly one attribute $c_i^j \in D^j$ for each domain $j \in [d]$. Let $a : C \to D$ be a function, which maps from a candidate c_i to her attribute vector $a(c_i) = (c_i^1, \dots, c_i^d)$.*
- *$V = \{v_1, \dots, v_n\}$ is a set of n voters, each $v_i \in V$ is associated with her ballot, represented as a vector $b_i = (B_i^1, \dots, B_i^d) \in \mathcal{D}$,*

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.

with $\mathcal{D} = \mathcal{P}(D^1)\times\ldots\times\mathcal{P}(D^d)$, i.e., each voter specifies which subset of attributes she approves of in each attribute category.

A voting rule F maps an election $E = (D, C, V) \in \mathcal{E}$ along with a positive integer $k \in \mathbb{N}_{>0}$ to a nonempty set of winning k-committees, i.e., $F(E, k) \subseteq \mathcal{P}_k(C)$. We assume that $|C| \geq k$ always holds.

We study voting rules, where the output is a set of k-committees that maximize the overall satisfaction of the voters. An *individual scoring function* f models any single voter's individual agreement (called satisfaction or score) of her attribute approval ballot $b_i \in \mathcal{D}$ with a given committee $W \subseteq C$, such that $f(b_i, W) \in \mathbb{Q}_{\geq 0}$. Examples for individual scoring functions are *Simple Scoring* (f^{si}), *Chamberlin-Courant Scoring* (f^{cc}), and *Committee Scoring* (f^{co}):

$$f^{\mathrm{si}}(b_i, W) = \frac{1}{d} \sum\nolimits_{c\in W} \sum\nolimits_{j\in[d]} |\{c^j\} \cap B_i^j|$$

$$f^{\mathrm{cc}}(b_i, W) = \frac{1}{d} \max_{c\in W} \sum\nolimits_{j\in[d]} |\{c^j\} \cap B_i^j|$$

$$f^{\mathrm{co}}(b_i, W) = \frac{1}{d} \left|\left\{ j \in [d] : \exists c \in W \text{ with } c^j \in B_i^j \right\}\right|$$

To obtain voting rules for attribute approval ballots we extend a given individual scoring function f from single ballots to *extended scoring functions* for voter profiles, considering two prominent approaches. Given a set V of voters and a k-committee W we either maximize the **utilitarian welfare** $f_\Sigma(V, W) = \sum_{v_i\in V} f(b_i, W)$ or the **egalitarian welfare** $f_{\min}(V, W) = \min_{v_i\in V} f(b_i, W)$. Finally, for each individual scoring function $f^y \in \{f^{\mathrm{si}}, f^{\mathrm{cc}}, f^{\mathrm{co}}\}$ paired with an extension $x \in \{\Sigma, \min\}$, a voting rule F_x^y maximizes the score of an extended scoring function f_x^y, outputting a set of winning k-committees, i.e., $F_x^y(E, k) = \arg\max_{W\in\mathcal{P}_k(C)} f_x^y(V, W)$. The modular setup provides six extended scoring functions and thus six voting rules[1] $F_\Sigma^{\mathrm{si}}, F_\Sigma^{\mathrm{cc}}, F_\Sigma^{\mathrm{co}}, F_{\min}^{\mathrm{si}}, F_{\min}^{\mathrm{cc}}$, and $F_{\min}^{\mathrm{co}}$.

Preference Derivation Methods. In many natural situations attribute approval ballots (in combination with a scoring function) model the underlying preferences realistically. If voters can only express their preferences in more common forms, it is reasonable to assume a voter either (i) assigns a utility to each candidate linear in the number of satisfied attributes, (ii) weakly ranks the candidates based on the number of satisfied attributes, or (iii) approves those candidates that satisfy a threshold amount of attributes.

Definition 2. *Let $E = (D, C, V) \in \mathcal{E}$ and $b_i = (B_i^1, \ldots, B_i^d)$ be the attribute approval ballot of voter $v_i \in V$. We study the following preference derivation methods for different types of preferences.*

- **Cardinal Preference Ballots:** $\mathfrak{c} : \mathcal{D}\times C \to \mathbb{Q}$, *where $\mathfrak{c}(b_i, c) = \frac{1}{d}|\{j \in [d] : c^j \in B_i^j\}|$ is the cardinal preference, voter v_i associates with candidate $c \in C$.*
- **Ordinal Preference Ballots:** $\mathfrak{o} : \mathcal{D} \to \mathcal{P}(C \times C)$, *such that $\mathfrak{o}(b_i) = \succsim_{b_i}$ is a weak ranking over C with $c \succ_{b_i} c'$ if $\mathfrak{c}(b_i, c) > \mathfrak{c}(b_i, c')$ and $c \sim_{b_i} c'$ if $\mathfrak{c}(b_i, c) = \mathfrak{c}(b_i, c')$ for all $c, c' \in C$.*
- **Candidate Approval Ballots:** $\mathfrak{a}_\tau : \mathcal{D} \to \mathcal{P}(C)$, *such that $\mathfrak{a}_\tau(b_i) = \{c \in C : \mathfrak{c}(b_i, c) \geq \frac{\tau}{d}\}$ is the set of preferred candidates for a given threshold $\tau \in [d]$.*

With $\Delta(E)$, for a preference derivation method $\Delta \in \{\mathfrak{c}, \mathfrak{o}, \mathfrak{a}_\tau\}$, we refer to the election (C, V'), where each voter $v_i \in V$ with attribute approval ballot b_i is substituted by $v_i' \in V'$ with derived ballot $\Delta(b_i)$.

[1] F_Σ^{si} has also been studied by Kagita et al. [9] under the name Approval Voting.

Table 1: Summary of our results on distortion for each scoring rule f_x^y paired with a derivation method Δ. Entry ∞ indicates unbounded distortion, while 1 indicates no distortion.

Δ	f_Σ^{si}	f_Σ^{cc}	f_Σ^{co}	$f_{\min}^{\mathrm{si}}$	$f_{\min}^{\mathrm{cc}}$	$f_{\min}^{\mathrm{co}}$
$\mathfrak{a}_\tau$	∞	∞	∞	∞	∞	∞
$\mathfrak{a}_1$	d	d	d	d	d	d
$\mathfrak{c}$	1	1	$\min(k, d)$	1	1	$\min(k, d)$
$\mathfrak{o}$	d	d	∞	∞	∞	∞

3 DISTORTION

In our setting distortion measures how much the voters' satisfaction can decline, if we derive the voters' preferences using a (possibly) less expressive method instead of attribute approvals. If the distortion is high, the potential upside for voting on attributes is huge. In contrast, if the distortion is low, there is no downside in voting on candidates directly. We are interested in the distortion associated with a preference derivation method and an extended scoring function. That is the maximum factor the satisfaction can be higher by considering attribute ballots instead of other common forms of ballots. In contrast to related work on distortion (see [12], [5] and [1]) we do not assume an underlying cardinal utility for each candidate, but that the attribute approvals capture the voters' opinions entirely.

Definition 3. *Let Δ be a preference derivation method, which maps from an attribute approval election (D, C, V) to an election (C, V'). Further, let f_x^y be an extended scoring function and $\sigma_\Delta : \mathcal{E} \to \mathcal{P}(\mathcal{E})$ be a function with $\sigma_\Delta(E) = \{E' \in \mathcal{E} : \Delta(E) = \Delta(E')\}$ for every $E \in \mathcal{E}$. That is, $\sigma_\Delta(E)$ is the set of attribute elections, that yield ballots equivalent to E if the votes are derived using Δ. In the following, for $\mathcal{E}' \subseteq \mathcal{E}$, let $\mathcal{W}(\mathcal{E}', k) = \bigcup_{E'\in\mathcal{E}'} F_x^y(E', k)$, be the collection of all winning k-committees for all elections in $\mathcal{E}'$. For a fixed attribute approval election $E = (D, C, V) \in \mathcal{E}$, the distortion associated with Δ, f_x^y, and E is given by*

$$\mathrm{dist}(\Delta, f_x^y, E) = \max_{W'\in\mathcal{W}(\sigma_\Delta(E),k)} \frac{\max_{W\in\mathcal{P}_k(C)} f_x^y(V, W)}{f_x^y(V, W')}$$

In case only the denominator is zero, we say the distortion is unbounded. The overall distortion for Δ and f_x^y (not depending on a specific election) is given by $\mathrm{dist}(\Delta, f_x^y) = \max_{E\in\mathcal{E}} \mathrm{dist}(\Delta, f_x^y, E)$.

The intuition for $\mathrm{dist}(\Delta, f_x^y, E)$ is, that E represents the undistorted voters' preferences, i.e., their attribute-based preferences. If the voters' preferences were instead cast by Δ with a loss of information (e.g., approval ballots), there is no way to determine which of the attribute approval elections in $\sigma_\Delta(E)$ coincides with the voters' actual ballots. If we pick an election $E' = (D', C, V') \in \sigma_\Delta(E)$, then a winning committee $W' \in F_x^y(E', k)$ maximizes the satisfaction for V'. Yet, the set of voters V might be dissatisfied with W', that is, $f_x^y(V, W')$ could be much lower than the score of a winning committee. We use a given extended scoring function f_x^y as a metric to evaluate the satisfaction of the voters with a committee. Our results are summarized in Table 1.

ACKNOWLEDGEMENTS

The authors would like to thank the anonymous reviewers for their helpful comments. This work was supported in part by the project "Online Participation," funded by the NRW Ministry for Innovation, Science, and Research and by the DFG grant BA-6270/1-1. Current affiliation of the first author: Federal University of Applied Administrative Science, Brühl, Germany.

REFERENCES

[1] E. Anshelevich, A. Filos-Ratsikas, N. Shah, and A. Voudouris. 2021. Distortion in Social Choice Problems: The First 15 Years and Beyond. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence*. ijcai.org, 4294–4301. Survey Track.

[2] N. Barrot and J. Lang. 2016. Conditional and Sequential Approval Voting on Combinatorial Domains. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence*. AAAI Press, 88–94.

[3] D. Baumeister, T. Böhnlein, L. Rey, O. Schaudt, and A. Selker. 2016. Minisum and Minimax Committee Election Rules for General Preference Types. In *Proceedings of the 22nd European Conference on Artificial Intelligence*. IOS Press, 1656–1657. Short paper.

[4] C. Boutilier, R. Brafman, C. Domshlak, H. Hoos, and D. Poole. 2004. CP-nets: A Tool for Representing and Reasoning with Conditional Ceteris Paribus Preference Statements. *Journal of Artificial Intelligence Research* 21 (2004), 135–191.

[5] Craig Boutilier, Ioannis Caragiannis, Simi Haber, Tyler Lu, Ariel D. Procaccia, and Or Sheffet. 2015. Optimal Social Choice Functions: A Utilitarian View. *Artificial Intelligence* 227 (2015), 190–213.

[6] R. Bredereck, P. Faliszewski, A. Igarashi, M. Lackner, and P. Skowron. 2018. Multiwinner Elections with Diversity Constraints. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*. AAAI Press, 933–940.

[7] P. Faliszewski, P. Skowron, A. Slinko, and N. Talmon. 2017. Multiwinner Voting: A New Challenge for Social Choice Theory. In *Trends in Computational Social Choice*, U. Endriss (Ed.). AI Access Foundation, 27–47.

[8] P. Jain and N. Talmon. 2020. Committee Selection with Multimodal Preferences. In *Proceedings of the 24nd European Conference on Artificial Intelligence*. IOS Press, 123–130.

[9] V. Kagita, A. Pujari, V. Padmanabhan, H. Aziz, and V. Kumar. 2021. Committee Selection using Attribute Approvals. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*. IFAAMAS, 683–691.

[10] J. Lang and P. Skowron. 2018. Multi-Attribute Proportional Representation. *Artificial Intelligence* 263 (2018), 74–106.

[11] S. Obraztsova, M. Polukarov, Z. Rabinovich, and E. Elkind. 2017. Doodle Poll Games. In *Proceedings of the 16th International Joint Conference on Autonomous Agents and Multiagent Systems*. ACM Press, 876–884.

[12] A. Procaccia and J. Rosenschein. 2006. The Distortion of Cardinal Preferences in Voting. In *Proceedings of the 10th International Workshop on Cooperative Information Agents*. Springer-Verlag *Lecture Notes in Artificial Intelligence #4149*, 317–331.

[13] W. Zwicker. 2016. Introduction to the Theory of Voting. In *Handbook of Computational Social Choice*, F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. Procaccia (Eds.). Cambridge University Press, Chapter 2, 23–56.

CHAPTER 10

BOUNDED APPROVAL BALLOTS: BALANCING EXPRESSIVENESS AND SIMPLICITY FOR MULTIWINNER ELECTIONS

In this chapter, we design and explore a novel ballot format and suitable scoring functions, allowing voters to express incompatibilities, dependencies, and substitution effects. In an extensive axiomatic analysis we evaluate the adequacy of our modelization and complement our findings by studying what kind of linear orders over committees may be induced by a ballot paired with a scoring function.

10.1 Summary

In this article, we introduce a new ballot format as a generalization of approval ballots, to allow voters to specify incompatibilities, dependencies, and substitution effects between alternatives. A formal description of the considered model can be found in Subsection 2.2.2. To recap informally, a bounded approval set is modeled by a set of approved candidates, accompanied by *(i)* a (numeric) lower bound to specify minimum candidates expected to be in the committee to receive any satisfaction, *(ii)* a saturation point to specify after how many candidates the satisfaction does not increase anymore, and *(iii)* an upper bound after which the satisfaction with a committee drops back to zero. Interpreting bounded sets accordingly, we consider a scoring function to quantify the agreement of a bounded approval set with a given committee as depicted above. Yet, to allow for more evolved interconnected preferences over the set of alternatives, a voter's bounded approval ballot may contain more than one bounded approval set. To measure the satisfaction of a voter with an outcome based on her ballot (possibly containing contradicting bounded approval sets), we distribute the score for each bounded set equally on all contributing

candidates. Finally, having a score per candidate calculated for each bounded set, the overall score for a candidate in a bounded approval ballot is gathered from considering either the minimum, maximum, average, or total satisfaction across all bounded approval sets. Additionally, we study a simple rule, which extends approval voting by assigning those candidates a unit score that are beneficial in at least one bounded set.

After briefly showing, that all resulting rules (selecting committees receiving a maximum score) are hard to compute, we continue with an extensive axiomatic analysis to evaluate the adequacy of our modelization. To meet our initial goals in mind, a voting rule should act just as the standard Approval Voting rule (AV) when possible, while incompatibilities, substitution effects, and dependencies should be taken into account when necessary. To satisfy the former, the score assigned to a ballot under a given committee should be *(i)* bounded upwards by the approval score for all alternatives appearing across all bounded sets, *(ii)* coincide with the approval score if no dependency, substitution, or incompatibility is present, and *(iii)* zero if and only if all bounded sets contain a violated dependency or incompatibility. In turn, to satisfy the latter, a violated dependency or incompatibility should not increase the score, while adding a substitute to a committee should not affect the score at all. Additionally, we consider desirable monotonicity axioms to evaluate the behavior associated with given scoring functions. Specifically, neither extending a bounded ballot with a non-conflicting bounded set, nor adding a suitable candidate, should decrease the score of a given committee. Likewise, expressing a bounded set in a (logically) equivalent statement using multiple ballots should not interfere with the score at all.

Overall, we establish a complete axiomatic analysis by demonstrating for each scoring function which axioms are either satisfied or violated. As none of our considered scoring functions accomplishes to satisfy all considered axioms, we subsequently were able to show that some of our axiomatic properties are indeed incompatible with one another. As a way to escape this impossibility result, we show that four of our scoring functions coincide in case bounded sets within a bounded ballot do not overlap. As a result, given this restriction, all axiomatic properties are satisfied simultaneously. Alternatively, we illustrate how weakening our axioms may also lead to almost fulfilling our modeling goals, by considering a rather artificial scoring function.

To complement our axiomatic results, we evaluate the expressiveness of our novel ballot format, coming from two different angles. On the one hand, we study the limits of what a voter's ballot can express under a given scoring function, by showing which kind of ordinal rankings over k-committees can be induced in theory (focusing on dichotomous, trichotomous, or arbitrary rankings). On the other hand, we present natural examples to showcase how the overall satisfaction may increase significantly by considering bounded ballots instead of approval ballots.

Lastly, to test our theoretical results empirically in real-world elections, we developed the web-application *GoodVotes*, which is now maintained under the name *GoodVoteX* [122].

10.2 Reflection on Initial Research Goals

In this article, we focused on four of our initial research questions, introduced in Chapter 3. Most obviously, we explored Question Q5 by introducing bounded approval ballots, as a way to extend standard approval ballots to capture incompatibilities, dependencies, and substitution effects. Particularly, we studied how expressive these types of ballots are compared to weak rankings over the set of all fixed-size committees, or to approval ballots. We significantly addressed Question Q1, as we initiated an extensive axiomatic analysis for bounded ballots. To escape an incompatibility result (showing that all initial modeling goals can not be met by a scoring function simultaneously), we demonstrated possible ways to weaken axioms or restrict the types of valid ballots slightly to satisfy all considered axioms. As an intermediate result, we showed that in case the bounded sets in a ballot do not overlap, four of our scoring rules coincide, contributing to Question Q4. We partially dealt with Question Q2, by designing reasonable scoring functions to model respective rules and briefly showing that all considered rules are computationally hard.

10.3 Publication

This work has been published and presented as a full paper at the *22nd International Conference on Autonomous Agents and Multiagent Systems*, along with the prototype application *GoodVotes*.

[23] D. Baumeister, L. Boes, C. Laußmann, and S. Rey. "Bounded Approval Ballots: Balancing Expressiveness and Simplicity for Multiwinner Elections". In: *Proceedings of the 22nd International Conference on Autonomous Agents and Multiagent Systems*. IFAAMAS, 2023, pp. 1400–1408

Parts of this article, containing more evolved insights into the *GoodVotes* web-application, were published in Christian Laußmann's dissertation [121].

10.4 Personal Contribution

This work was initiated by Simon Rey's research stay at Düsseldorf in early 2022. The design and formalization of the ballot format and suitable scoring functions was developed in productive meetings by all authors in equal parts. Similarly, the conception and writing of the article was done by an equal share of work by all participating authors. As for the technical results: The axiomatic analysis (Section 3) was mostly conducted by Christian Laußmann and Simon Rey, the comparison to approval ballots (Section 4.2) was analyzed by Christian Laußmann, while the study on the limits of expressiveness (Section 4.1) was primarily contributed by me. The preliminary web-application *GoodVotes* was designed and developed by Christian Laußmann.

Bounded Approval Ballots: Balancing Expressiveness and Simplicity for Multiwinner Elections

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

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

Christian Laußmann
Heinrich-Heine-Universität
Düsseldorf, Germany
christian.laussmann@uni-duesseldorf.de

Simon Rey
ILLC, University of Amsterdam
Amsterdam, The Netherlands
s.j.rey@uva.nl

ABSTRACT

Approval ballots have been celebrated for many voting scenarios [16], in particular because of the low cognitive burden they put on the voters. This however, comes at the cost of expressiveness that can be problematic when voters have sophisticated preferences. We consider voters who, in addition to usual approval, may wish to express incompatibilities, dependencies, and/or substitution effects between the alternatives. We introduce, and evaluate a new type of ballot—*bounded approval ballots*—which captures these effects while being almost as easy as regular approval ballots to cast.

KEYWORDS

Multiwinner Voting; Approval Voting; Ballot Design

ACM Reference Format:
Dorothea Baumeister, Linus Boes, Christian Laußmann, and Simon Rey. 2023. Bounded Approval Ballots: Balancing Expressiveness and Simplicity for Multiwinner Elections. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 9 pages.

1 INTRODUCTION

Let us focus on the case study of *Goodman's Pipes and Tubes Ltd.*, a company that is about to elect its expert committee. The committee consists of four people who advise the board on business strategy questions. The following six candidates are up for election: *Anna* and *Chris* are two of the leading engineers in the company; *Ben* from the human resources; *Diana* from the legal department; *Elena* from the advertisement department; the external craftsman *Frank*; and *Gustavo* who is responsible for material purchases. The three board members *Rob*, *Su*, and *Tim* have the following opinions.

Rob: "I'm happy with Anna's, Chris', Elena's, Frank's, and Gustavo's work. They are reliable, work hard, and have been around long enough, so each of them will improve the committee with their own expertise. "

Su: "We are an engineering company. Expertise on materials, production, and craftsmanship should be our focus, so there should be one, or better two of Anna, Chris, Frank, and Gustavo. Having more of them is also fine, although I don't see a big advantage for that. I think there should also be expertise from Diana in the committee. However, Diana and Chris shouldn't be together because they argue a lot. "

Tim: "I am pretty sure that Chris, Gustavo, Ben, and Diana will do a good job. However, every time I discuss something with either Ben or Diana, the other one is angry because they say that the legal department and the human resources must work closely together. So if we give one of them a seat in the committee, the other one must also get a seat, or else we can just as well give no seat to either of the two. "

The committee selection example above is a canonical instance of *multiwinner elections* [11]. In this context, the most common way of asking one's opinion is to use approval ballots: the voters indicate which of the alternatives they approve of [15]. Rob would for instance express his preference by approving of Anna, Chris, Elena, Frank, and Gustavo. However, Su's and Tim's statements cannot be expressed as approval ballots. It is true that Su in principle approves of Anna, Chris, Diana, Frank, and Gustavo. However, by using an approval ballot she cannot state that two of them are just as good as three, nor that Diana and Chris are incompatible. Similarly, Tim cannot express in an approval ballot that either Ben and Diana must be in the committee, or neither of them. Our goal is to find a ballot format to account for the type of preferences illustrated above, *i.e.*, approvals, incompatibilities, substitutions, and dependencies. Obviously, very expressive ballot formats (*e.g.* rankings over subsets of alternatives) could be used to express those, and even more complicated, preferences. This approach is however not satisfactory. Indeed, we believe that more expressive ballots should still be practical, *i.e.*, not imposing a high cognitive burden on the voters, and scaling reasonably well as the number of alternatives increases. Moreover, even though some voters can be interested in submitting complex ballots, only proposing complex ballots can prevent some others to participate. We thus want to develop ballot formats that still allow for simple ballots to be submitted.

Contribution. To achieve the goals described above, we introduce *bounded approval ballots.* A bounded approval ballot is a collection of bounded approval sets: sets of approved alternatives that are enriched with three bounds: a *lower bound* (minimum number of alternatives that have to be selected), a *saturation point* (number of selected alternatives after which no additional satisfaction is derived), and an *upper bound* (maximum number of alternatives

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.

that should be selected). Approval ballots are still valid (with simple reformatting) and treated exactly as in usual multiwinner approval voting to be convenient for the voters who don't want to take the effort to submit a more complicated ballot. Moreover, for voters who want to submit more sophisticated ballots (incompatibilities, substitutions and dependencies) the cognitive burden is no higher than setting some bounds for the ballot.

Related Work. Multiwinner voting (as a special case of voting in combinatorial domains [8]) has become a widely studied research area over the past years. We refer to Faliszewski et al. [11] for an overview of multiwinner voting rules and typical applications. Interestingly, the two most often used ballot types are approval ballots and ordinal ballots (rankings). Most of the research in the field focuses on the development of voting rules for such ballots (for example to guarantee fairness [1, 7]) rather than on the design and the study of these ballot types. Closest to what we are trying to achieve here are conditional preferences, where the preferences of a voter are conditioned on the status of a given variable. Several proposals have been discussed to express those opinions, the most famous probably being *conditional approval ballots* [3], *conditional preference networks* [5], and *lexicographic preference trees* [4]. A stream of research for combinatorial auctions focuses on modeling complex utility functions using expressive languages [9]. Sandholm [18] studies bidding languages, where atomic bids are joined with logical connectives (allowing for substitution and incompatibility effects), while Hoos and Boutilier [6, 13] interconnect bids and logical formulas over the alternatives. However, all these bidding languages are rather complicated to express, at least compared to the very simple approval ballot format. Finally, several proposals have been made to extend approval ballots, mainly in the context of participatory budgeting, a generalization of multiwinner elections [2]. Jain et al. [14] partition the alternatives into categories to model interactions among them. Fairstein et al. [10] incorporate individual partitions of the alternatives to study substitution effects (in a slightly different way than we do).

2 PRELIMINARIES

A multiwinner election consists of a set of m alternatives (also called candidates) $\mathcal{A} = \{a_1, \ldots, a_m\}$, a profile $\mathfrak{B} = (\boldsymbol{B}_1, \ldots, \boldsymbol{B}_n)$ which is a list of ballots $\boldsymbol{B}_i$ of n voters $\mathcal{N} = \{1, \ldots, n\}$, and an integer $k \in \{1, \ldots, m\}$. We denote by $C_k = \{\pi \subseteq \mathcal{A} \mid |\pi| = k\}$ the set of all k-sized committees. The outcome of an irresolute multiwinner election is a set of winning committees $\{\pi_1, \pi_2, \ldots\} \subseteq C_k$. The ballot format is described in the next section. We will use $\oplus$ to denote the concatenation operator between two lists. The subtraction of list B from list A will be denoted through $A \ominus B$ (where for each element in B the first occurrence of the element in A is removed). We sometimes omit the brackets around a list of length one.

2.1 Bounded Approval Ballots

We now introduce bounded approval ballots, our generalization of approval ballots to allow for submitting more complex preferences.

Definition 1 (Bounded Sets and Ballots)**.** *Given a set of alternatives $\mathcal{A}$, a* bounded (approval) set *is a tuple $B^j = \langle A^j, \ell^j, s^j, u^j \rangle$ such that $A^j \subseteq \mathcal{A}$ and ℓ^j, s^j, u^j, respectively the lower bound, the saturation point, and the upper bound, are all integers such that $1 \leq \ell^j \leq s^j \leq u^j \leq |A^j|$. A* bounded (approval) ballot *$\boldsymbol{B}_i$, for voter $i \in \mathcal{N}$, is a list $\boldsymbol{B}_i = (B_i^1, \ldots, B_i^p)$ of bounded sets.*

A bounded set indicates that from all the alternatives in A^j, at least ℓ^j but no more than u^j have to be selected; while after s^j alternatives have been selected from A^j, the voter will not enjoy any additional satisfaction from selecting more alternatives.[1] This way, we achieve all of our initial modeling goals:

- *Standard approval ballots* can be expressed by setting $\ell^j = 1$, $s^j = u^j = |A^j|$: the more alternatives from A^j the better;
- *Incompatibilities* can be expressed by bounded sets with an upper bound $u^j = 1$: Selecting multiple alternatives from A^j is not desired by the voter because these alternatives are incompatible, but selecting one is desirable;
- *Substitution* can be expressed by bounded sets with $\ell^j = s^j = 1$ and $u^j = |A^j|$: Selecting one alternative from A^j is desired but additional alternatives are substitutes;
- *Dependencies* can be expressed by bounded sets where $\ell^j = |A^j|$: All alternatives from A^j rely on each other, and are only useful for the voter if all of them are present.

We illustrate these ballots on the example from the introduction.

Example 1. Rob only wants to provide an approval ballot that can be expressed by a simple bounded approval ballot consisting of only one bounded set: $\langle$\{Anna, Chris, Elena, Frank, Gustavo\}, 1, 5, 5$\rangle$.

Su's preference is more involved. The incompatibility between Diana and Chris can be expressed by $\langle$\{Diana, Chris\}, 1, 1, 1$\rangle$. Further, with $\langle$\{Anna, Chris, Frank, Gustavo\}, 1, 2, 4$\rangle$, we can express that one—or better two—of Anna, Chris, Frank, and Gustavo should be included but there is no further benefit for three or four. Su's ballot would thus consist of these two bounded sets.

Tim states that from Ben and Diana either both or none should be included, which can be expressed as $\langle$\{Ben, Diana\}, 2, 2, 2$\rangle$. Furthermore, both Chris and Gustavo are approved, which can be expressed as $\langle$\{Chris, Gustavo\}, 1, 2, 2$\rangle$. △

Finally, we introduce one useful notation: for a ballot $\boldsymbol{B}$ and an alternative $a \in \mathcal{A}$, we denote by $\boldsymbol{B}_{|a} = \{B^j \in \boldsymbol{B} \mid a \in A^j\}$ the bounded sets in $\boldsymbol{B}$ involving a.

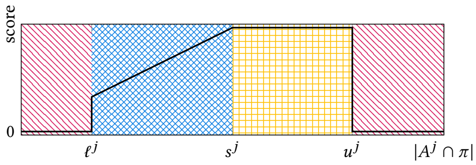
2.2 Scoring with Bounded Approval Ballots

We eventually want to aggregate the ballots that the voters submitted in order to determine a winning committee. In the following we provide different *scoring functions* which map profiles and committees to real numbers. These can then be used to define rules by simply selecting the committee with the highest score. We will investigate properties of the scoring functions later.

Definition 2 (Scoring Function)**.** *A scoring function score is a function taking as input a bounded approval ballot $\boldsymbol{B}$ and a committee π, and returning a real value score$(\boldsymbol{B}, \pi)$. We extend scoring functions to profiles s.t. for every profile $\mathfrak{B}$, score$(\mathfrak{B}, \pi) = \sum_{\boldsymbol{B} \in \mathfrak{B}}$ score$(\boldsymbol{B}, \pi)$.*

To capture the semantics of bounded sets described above, for a committee π and a bounded set $B^j = \langle A^j, \ell^j, s^j, u^j \rangle$, we want scoring functions to behave as depicted below.

[1] We assume that all alternatives in A^j are *approved* in the sense that for each $a \in A^j$ there is a committee in which the voter would like a to be part of.



The area represents that π violates a dependency or incompatibility stated in B^j. Thus, π should score 0. The area represents that each element in π is independently approved according to B^j—each contributes to the total score. Finally, the area represents that the elements in π show substitution effects according to B^j—additional items contribute no additional score.

Assuming neutrality, *i.e.,* that all alternatives are treated the same, we introduce a function that should be interpreted as the average score of the alternatives appearing in B^j. When within the lower bound and the saturation point, each alternative from $A^j \cap \pi$ fully contributes to B^j's score, *i.e.,* they score 1 each. If the saturation point is exceeded, the s^j points are equally split among the alternatives. If the lower or the upper bound is violated, all the alternatives score 0. The formal definition of this function φ is:

$$\varphi(B^j, \pi) = \begin{cases} 1 & \text{if } \ell^j \leq |A^j \cap \pi| \leq s^j \\ \frac{s^j}{|A^j \cap \pi|} & \text{if } s^j < |A^j \cap \pi| \leq u^j \\ 0 & \text{otherwise.} \end{cases}$$

Example 2. Consider the following bounded set from Su's ballot: $B^j = \langle\{\text{Anna, Chris, Frank, Gustavo}\}, 1, 2, 4\rangle$. For $\pi = \{\text{Anna, Ben, Chris, Diana}\}$ we have $\varphi(B^j, \pi) = 1$. That is, according to B^j each alternative in $\pi \cap A^j$ is fully approved, so in total π has two approvals—one for Anna and one for Chris. For $\pi' = \{\text{Anna, Ben, Chris, Frank}\}$ the saturation bound is exceeded, so we have $\varphi(B^j, \pi') = \frac{2}{3}$. △

Scoring functions are defined for bounded approval ballots (and profiles of them), and not for bounded sets as φ is defined. If each ballot consists of only one bounded set, this would be straightforward. However, as soon as there are several bounded sets, we need to aggregate the score of the individual bounded sets. Several usual operators can be considered here: averaging, taking the minimal or the maximal value, or simply summing up the scores. This results in the following scoring functions, all based on φ.

$$score_{min}(\boldsymbol{B}, \pi) = \sum_{a \in \pi} \min\left(\varphi(B^j, \pi) \mid B^j \in \boldsymbol{B}_{|a}\right)$$

$$score_{max}(\boldsymbol{B}, \pi) = \sum_{a \in \pi} \max\left(\varphi(B^j, \pi) \mid B^j \in \boldsymbol{B}_{|a}\right)$$

$$score_{avg}(\boldsymbol{B}, \pi) = \sum_{a \in \pi} \frac{1}{|\boldsymbol{B}_{|a}|} \sum_{B^j \in \boldsymbol{B}_{|a}} \varphi(B^j, \pi)$$

$$score_{tot}(\boldsymbol{B}, \pi) = \sum_{a \in \pi} \sum_{B^j \in \boldsymbol{B}_{|a}} \varphi(B^j, \pi)$$

Note that the semantics we developed is respected when each ballot consists of a single bounded set. Note further, that the functions coincide when each alternative is part of at most one bounded set per ballot, *i.e.*, when $|\boldsymbol{B}_{|a}| \leq 1$ for all $a \in \mathcal{A}$ and every ballot $\boldsymbol{B}$.

In addition to these four scoring functions, we also study another one that is not based on φ: $score_{app}$. It is a natural generalization of the approval score, as it counts the number of alternatives in π for which there exists at least one bounded set B^j for which the lower bound is exceeded in π, but not the saturation point:

$$score_{app}(\boldsymbol{B}, \pi) = |\{a \in \pi \mid \exists B^j \in \boldsymbol{B}_{|a} \text{ s.t. } \ell^j \leq |A^j \cap \pi| \leq s^j\}|.$$

Since, $score_{app}$ completely disregards substitution effects, it will not exactly fit the framework we detailed above. It will be shown to provide interesting axiomatic results still.

Before moving on to the axiomatic analysis, let us briefly discuss the computational complexity of the scoring functions. For all of them, finding a committee with maximal score cannot be done in polynomial-time, unless P = NP. For the φ-based rules it is easy to see that we can simulate the (approval version of the) Chamberlin–Courant rule with them by submitting a single bounded set per voter, where each bounded set has a saturation point of one and an upper bound involving all approved alternatives. The observation then follows from the fact that Chamberlin–Courant winner determination is NP-hard [19]. In the case of $score_{app}$, we can use the NP-hard problem Exact Cover by 3-Sets (see Garey and Johnson [12]) to show the claim. These downsides are, unfortunately, unavoidable when working with more expressive ballot formats.

3 ADEQUACY OF THE MODELIZATION

Following the classical method of social choice [20], we develop an axiomatic theory to investigate the behavior of scoring functions.

3.1 Axiomatic Theory

We encode, by the means of axioms, the idea that bounded approval ballots allow voters to express the different statements we are interested in; and that the scoring functions comply with the semantics we are aiming for.

We first define two axioms enforcing that a violated incompatibility or dependency should not increase the score.

Definition 3 (Incompatibility Adequacy)**.** *A scoring function score satisfies* incompatibility adequacy *if for every $A \subseteq \mathcal{A}$, and all ballots $\boldsymbol{B}$ and $\boldsymbol{B}' = \boldsymbol{B} \oplus \langle A, 1, 1, 1\rangle$, the following holds:*

- *$score(\boldsymbol{B}, \pi) \leq score(\boldsymbol{B}', \pi)$ for every π with $|\pi \cap A| = 1$;*
- *$score(\boldsymbol{B}, \pi) \geq score(\boldsymbol{B}', \pi)$ for every π with $|\pi \cap A| \neq 1$.*

Definition 4 (Dependency Adequacy)**.** *A scoring function score satisfies* dependency adequacy *if for every $A \subseteq \mathcal{A}$, and all ballots $\boldsymbol{B}$ and $\boldsymbol{B}' = \boldsymbol{B} \oplus \langle A, |A|, |A|, |A|\rangle$, the following holds:*

- *$score(\boldsymbol{B}, \pi) \leq score(\boldsymbol{B}', \pi)$ for every π with $A \subseteq \pi$;*
- *$score(\boldsymbol{B}, \pi) \geq score(\boldsymbol{B}', \pi)$ for every π with $A \not\subseteq \pi$.*

Our next axiom concerns properly modeling substitution. Informally, if according to all bounded sets an item $a^\star$ is considered a substitute w.r.t. π, then adding $a^\star$ to π should not change the score.

Definition 5 (Substitution Adequacy)**.** *A scoring function score satisfies* substitution adequacy *if for every ballot $\boldsymbol{B}$ and committee π for which there exists an alternative $a^\star \in \mathcal{A} \setminus \pi$ such that for all bounded sets $B^j \in \boldsymbol{B}_{|a^\star}$, it is the case that $s^j \leq |A^j \cap \pi| \leq u^j - 1$, we have $score(\boldsymbol{B}, \pi) = score(\boldsymbol{B}, \pi \cup \{a^\star\})$.*

Next, we ensure that a scoring function treats approval ballots correctly, *i.e.*, that it behaves as the usual approval score for standard approval ballots.

Definition 6 (Approval Adequacy). *A scoring function score satisfies* approval adequacy *if for every ballot $\boldsymbol{B}$ and committee π the following two conditions hold:*

(1) $score(\boldsymbol{B}, \pi) \leq |\left(\bigcup_{B^j \in \boldsymbol{B}} A^j\right) \cap \pi|$;

(2) $score(\boldsymbol{B}, \pi) = |\left(\bigcup_{B^j \in \boldsymbol{B}} A^j\right) \cap \pi|$ *whenever* $\ell^j \leq |A^j \cap \pi| \leq s^j$ *for all* $B^j \in \boldsymbol{B}$.

The final adequacy axiom requires the scoring function to return 0 if and only if there is a good reason to do so.

Definition 7 (Zero Adequacy). *A scoring function score satisfies* zero-adequacy *if for every ballot $\boldsymbol{B}$ and committee π we have:*

$$score(\boldsymbol{B}, \pi) = 0 \quad \textit{iff} \quad \forall B^j \in \boldsymbol{B}, |A^j \cap \pi| > u^j \text{ or } |A^j \cap \pi| < \ell^j.$$

We further introduce monotonicity axioms enforcing the scoring rules to be well-behaved in a dynamic environment.

The first one says that adding a bounded set which does not conflict with a committee π should not decrease the score of π.

Definition 8 (Ballot-Size Monotonicity). *Let $\boldsymbol{B}$ be a ballot and π a committee. A scoring function score satisfies* ballot-size monotonicity *if for every bounded set $B = \langle A, \ell, s, u\rangle$ such that $\ell \leq |A \cap \pi| \leq u$, we have $score(\boldsymbol{B}, \pi) \leq score(\boldsymbol{B} \oplus B, \pi)$.*

Ballot-splitting monotonicity says that expressing an equivalent statement with one large ballot, or several smaller ones, should result in the same score.

Definition 9 (Ballot-Splitting Monotonicity). *A scoring function score satisfies* ballot-splitting monotonicity *if for every committee π and every ballot $\boldsymbol{B}$ for which there exists a bounded set $B^{j^\star} \in \boldsymbol{B}$ such that $\ell^{j^\star} \leq |A^{j^\star} \cap \pi| \leq s^{j^\star}$, then, for $\boldsymbol{B}' = (\boldsymbol{B} \ominus B^{j^\star}) \oplus (\langle \{a\}, 1, 1, 1\rangle \mid a \in A^{j^\star} \cap \pi)$, we must have $score(\boldsymbol{B}, \pi) = score(\boldsymbol{B}', \pi)$.*

Finally, score monotonicity requires the score not to decrease when adding a suitable alternative to the committee.

Definition 10 (Score Monotonicity). *A scoring function score satisfies* score monotonicity *if for every ballot $\boldsymbol{B}$ and committee π for which there exists an alternative $a^\star \in \mathcal{A} \setminus \pi$ such that for all bounded sets $B^j \in \boldsymbol{B}_{|a^\star}$ it is the case that $\ell^j \leq |A^j \cap \pi| \leq u^j - 1$, we have that $score(\boldsymbol{B}, \pi) \leq score(\boldsymbol{B}, \pi \cup \{a^\star\})$.*

3.2 Axiomatic Behavior of Scoring Functions

Now that we have introduced a complete axiomatic theory, we investigate the performance of the scoring functions we introduced regarding those axioms. We start with $score_{avg}$.

Theorem 3. *The scoring function $score_{avg}$ satisfies approval, incompatibility, dependency, and zero adequacy, as well as ballot-splitting monotonicity. It fails ballot-size monotonicity, score monotonicity, and substitution adequacy.*

Proof. Let $\boldsymbol{B}$ be a ballot and π a committee.

Approval Adequacy (✓) For every alternative $a \in \pi$, $score_{avg}$ scores the average fulfillment of the relevant bounded sets. The fulfillment being a number between 0 and 1, the average also is between 0 and 1. We thus have $score_{avg}(\boldsymbol{B}, \pi) \leq |\left(\bigcup_{B^j \in \boldsymbol{B}} A^j\right) \cap \pi|$.

Now assume that $\ell^j \leq |A^j \cap \pi| \leq s^j$ for all $B^j \in \boldsymbol{B}$. Then, for all $B \in \boldsymbol{B}$, we have $\varphi(B, \pi) = 1$. Each alternative in π then scores 1, meaning that $score_{avg}(\boldsymbol{B}, \pi) = |\left(\bigcup_{B^j \in \boldsymbol{B}} A^j\right) \cap \pi|$. ∘

Substitution Adequacy (✗) Let $\mathcal{A} = \{a_1, a_2, a_3\}$, $\pi = \{a_1, a_2\}$, and $\boldsymbol{B} = (\langle\{a_1, a_2\}, 1, 2, 2\rangle, \langle\{a_1, a_3\}, 1, 1, 2\rangle)$. Note that a_3 fulfills the conditions required for $a^\star$ in the definition of substitution adequacy (Definition 5). On the one hand, we have $score_{avg}(\boldsymbol{B}, \pi) = 2/2 + 1/1 = 2$. On the other hand, for $\pi' = \{a_1, a_2, a_3\}$, we have $score_{avg}(\boldsymbol{B}, \pi') = \frac{1+1/2}{2} + 1/1 + \frac{1/2}{1} = 9/4 > 2$. ∘

Incompatibility Adequacy (✓) Let $\boldsymbol{B}' = \boldsymbol{B} \oplus \langle A, 1, 1, 1\rangle$ be the ballot with added incompatibility.

First assume $|\pi \cap A| \neq 1$. It is clear that for each $a \in A$, $|\boldsymbol{B}'_{|a}| = 1 + |\boldsymbol{B}_{|a}| > |\boldsymbol{B}_{|a}|$ holds, *i.e.*, the normalization factor decreases. This decrement together with $\varphi(\langle A, 1, 1, 1\rangle, \pi) = 0$ results in a decreased score contribution of a and thus $score_{avg}(\boldsymbol{B}, \pi) > score(\boldsymbol{B}', \pi)$.

Now assume that $\pi \cap A = \{a\}$ for some $a \in \mathcal{A}$. We distinguish three cases. (1) If $\boldsymbol{B}_{|a} = \emptyset$, then $score(\boldsymbol{B}', \pi) = score_{avg}(\boldsymbol{B}, \pi) + 1 > score_{avg}(\boldsymbol{B}, \pi)$. (2) If $\boldsymbol{B}_{|a} \neq \emptyset$ and for all $B^j \in \boldsymbol{B}_{|a}$, $\ell^j \leq |A^j \cap \pi| \leq s^j$ holds. Then clearly $score_{avg}(\boldsymbol{B}, \pi) = score(\boldsymbol{B}', \pi)$. (3) Finally, assume $\boldsymbol{B}_{|a} \neq \emptyset$ but for some $B^j \in \boldsymbol{B}_{|a}$ either $|A^j \cap \pi| < \ell^j$ or $s^j < |A^j \cap \pi|$. Let $\alpha = \sum_{B^j \in \boldsymbol{B}_{|a}} \varphi(B^j, \pi)$ and $\beta = |\boldsymbol{B}_{|a}|$. By assumption, $\alpha < \beta$ holds. Note that α/β is the contribution of a to the score of ballot $\boldsymbol{B}$. Moreover, $\alpha{+}1/\beta{+}1$ is the contribution of a to the score of $\boldsymbol{B}'$. Since for any $0 \leq \alpha < \beta$ we have $\alpha/\beta < \alpha{+}1/\beta{+}1$, we immediately obtain $score_{avg}(\boldsymbol{B}, \pi) < score(\boldsymbol{B}', \pi)$. ∘

Dependency Adequacy (✓) Let $\boldsymbol{B}' = \boldsymbol{B} \oplus \langle A, |A|, |A|, |A|\rangle$ be the ballot with added dependency.

First, assume $A \not\subseteq \pi$. Trivially, if A is disjoint with the other bounded sets, then $score_{avg}(\boldsymbol{B}, \pi) = score(\boldsymbol{B}', \pi)$. Otherwise, note that for each $a \in A$ that also occurs in another bounded set, we have $|\boldsymbol{B}'_{|a}| = 1 + |\boldsymbol{B}_{|a}| > |\boldsymbol{B}_{|a}|$. The normalization factor thus decreases, and since $\varphi(\langle A, |A|, |A|, |A|\rangle, \pi) = 0$, the contribution of a to the score also decreases. Overall, $score_{avg}(\boldsymbol{B}, \pi) > score(\boldsymbol{B}', \pi)$ holds.

Now assume $A \subseteq \pi$. For each element $a \in A$ we distinguish three cases. (1) If $\boldsymbol{B}_{|a} = \emptyset$, then clearly a increases the total score by 1. (2) If $\boldsymbol{B}_{|a} \neq \emptyset$ and for all $B^j \in \boldsymbol{B}_{|a}$, $\ell^j \leq |A^j \cap \pi| \leq s^j$ holds, then a contributed 1 to the total score in $\boldsymbol{B}$ and also in $\boldsymbol{B}'$ so nothing changes. (3) Finally, assume $\boldsymbol{B}_{|a} \neq \emptyset$ but for some $B^j \in \boldsymbol{B}_{|a}$ either $|A^j \cap \pi| < \ell^j$ or $s^j < |A^j \cap \pi|$. Let $\alpha = \sum_{B^j \in \boldsymbol{B}_{|a}} \varphi(B^j, \pi)$ and $\beta = |\boldsymbol{B}_{|a}|$. By assumption, $\alpha < \beta$ holds. Note that α/β is the contribution of a to the score of ballot $\boldsymbol{B}$. Moreover, $\alpha{+}1/\beta{+}1$ is the contribution of a to the score of $\boldsymbol{B}'$. Since for any $0 \leq \alpha < \beta$ we have $\alpha/\beta < \alpha{+}1/\beta{+}1$, we have $score_{avg}(\boldsymbol{B}, \pi) < score(\boldsymbol{B}', \pi)$. ∘

Zero Adequacy (✓) Note that we always have $score_{avg}(\boldsymbol{B}, \pi) \geq 0$. Moreover, $score_{avg}(\boldsymbol{B}, \pi) = 0$ iff $\varphi(B^j, \pi) = 0$ for all $B^j \in \boldsymbol{B}$, which holds iff for all $B^j \in \boldsymbol{B}$, either $|A^j \cap \pi| > u^j$ or $|A^j \cap \pi| < \ell^j$. ∘

Ballot-Size Monotonicity (✗) Let $\mathcal{A} = \{a_1, a_2, a_3\}$, $\pi = \{a_1, a_2\}$, and $\boldsymbol{B} = (\langle\{a_1, a_2\}, 1, 2, 2\rangle)$. Observe that we have $score_{avg}(\boldsymbol{B}, \pi) = 1/1 + 1/1 = 2$. Now, let $\boldsymbol{B}' = \boldsymbol{B} \oplus \langle\{a_1, a_2, a_3\}, 1, 1, 3\rangle$. We would then get $score_{avg}(\boldsymbol{B}', \pi) = \frac{1+1/2}{2} + \frac{1+1/2}{2} = 3/2 < 2$. This shows that ballot-size monotonicity is not satisfied. ∘

Ballot-Splitting Monotonicity (✓) Assume $B^{j^\star} \in \boldsymbol{B}$ is a set with $\ell^{j^\star} \leq |A^{j^\star} \cap \pi| \leq s^{j^\star}$. Let $\boldsymbol{B}' = (\boldsymbol{B} \ominus B^{j^\star}) \oplus (\langle\{a\}, 1, 1, 1\rangle \mid a \in A^{j^\star} \cap \pi)$. Note that $|\boldsymbol{B}_{|a}| = |\boldsymbol{B}'_{|a}|$ for all $a \in \pi$. So overall splitting the ballot has no effect on the normalization factor. Now, it is clear from the assumption that $\varphi(B^{j^\star}, \pi) = 1$. In $\boldsymbol{B}'$, for each of the newly created bounded set $B'^{j^\star}_i$, we have $\varphi(B'^{j^\star}_i, \pi) = 1$, too. Overall, nothing changes and $score_{avg}$ satisfies splitting monotonicity. ◦

Score Monotonicity (✗) Let $\mathcal{A} = \{a_1, a_2, a_3\}$, $\pi = \{a_2, a_3\}$, and $\boldsymbol{B} = (\langle\{a_1, a_2\}, 1, 1, 2\rangle, \langle\{a_1, a_3\}, 1, 1, 2\rangle)$. Note that a_1 fulfills the conditions required for $a^\star$ in the definition of score monotonicity (Definition 10). We have $score_{avg}(\boldsymbol{B}, \pi) = 1/1 + 1/1 = 2$. However, $score_{avg}(\boldsymbol{B}, \pi \cup \{a_1\}) = \frac{1/2+1/2}{2} + 1/2 + 1/2 < 2$. This shows that score monotonicity is not satisfied. □

Interestingly, the $score_{avg}$ scoring function fails substitution adequacy. As we shall see, this is due to its normalization factor. Indeed, $score_{tot}$ will not suffer this drawback, but suffers some others.

Theorem 4. *The scoring function $score_{tot}$ satisfies substitution, incompatibility, dependency, and zero adequacy, as well as ballot-size monotonicity, score monotonicity, and ballot-splitting monotonicity, but it fails approval adequacy.*

Proof. Let $\boldsymbol{B}$ be a ballot and π a committee.

Approval Adequacy (✗) Let $\mathcal{A} = \{a_1, a_2\}$, $\pi = \{a_1\}$, and $\boldsymbol{B} = (\langle\{a_1\}, 1, 1, 1\rangle, \langle\{a_1, a_2\}, 1, 2, 2\rangle)$. We have $score_{tot}(\boldsymbol{B}, \pi) = 2$ which is a clear violation of approval adequacy. ◦

Substitution Adequacy (✓) Consider a bounded approval ballot $\boldsymbol{B}$, a committee π and an alternative $a^\star \in \mathcal{A} \setminus \pi$ as in the definition of substitution adequacy (Definition 5). Let $B = \langle A, \ell, s, u\rangle$ be an arbitrary bounded set from $\boldsymbol{B}$ such that $a^\star \in A$. By the definition of $a^\star$, we know that $s \leq |A \cap \pi| \leq u - 1$. Hence, the contribution of B to $score_{tot}(\boldsymbol{B}, \pi)$ is s^j. Now, for $\pi' = \pi \cup \{a^\star\}$ we have $s + 1 \leq |A \cap \pi'| \leq u$. Hence, the contribution of B to $score_{tot}(\boldsymbol{B}, \pi')$ is also s^j. This applies to any bounded set including $a^\star$. Since the contributions of sets which don't include $a^\star$ are also unchanged, we have $score_{tot}(\boldsymbol{B}, \pi) = score_{tot}(\boldsymbol{B}, \pi')$. ◦

Incompatibility Adequacy (✓) Note that by adding a bounded set to a ballot the score cannot decrease. Further, if for the added ballot B^j, it holds that $u^j < |\pi \cap A^j|$, the score does not increase. ◦

Dependency Adequacy (✓) Note that by adding a bounded set to a ballot the score cannot decrease. Further, if for the added ballot B^j holds $\ell^j > |\pi \cap A^j|$, the score also does not increase. ◦

Zero Adequacy (✓) Note that we always have $score_{tot}(\boldsymbol{B}, \pi) \geq 0$. Moreover, $score_{tot}(\boldsymbol{B}, \pi) = 0$ iff $\varphi(B^j, \pi) = 0$ for all $B^j \in \boldsymbol{B}$, which holds iff for all $B^j \in \boldsymbol{B}$, either $|A^j \cap \pi| > u^j$ or $|A^j \cap \pi| < \ell^j$. ◦

Ballot-Size Monotonicity (✓) Note that by adding a bounded set B^j to a ballot the score cannot decrease. If for a committee π holds $\ell^j \leq |A^j \cap \pi| \leq u^j$, the score will even strictly increase. Thus, $score_{tot}$ satisfies ballot-size monotonicity. ◦

Ballot-Splitting Monotonicity (✓) Note that replacing B^j with $\ell^j \leq |A^j \cap \pi| \leq s^j$ by the bounded sets $(\langle\{a\}, 1, 1, 1\rangle \mid a \in A^j \cap \pi)$ means replacing a bounded set which contributes $|A^j \cap \pi|$ to the total score by $|A^j \cap \pi|$ many bounded sets which contribute 1 to the total score (w.r.t. π). This means no change for the score, *i.e.*, ballot-splitting monotonicity is satisfied. ◦

Score Monotonicity (✓) Let $a^\star$ be the alternative described in the definition. For bounded sets B^j with $\ell^j \leq |A^j \cap \pi| \leq s^j - 1$ it is immediate that the score contributions of alternatives in $A^j \cap \pi$ are the same in $score_{tot}(\boldsymbol{B}, \pi)$ and $score_{tot}(\boldsymbol{B}, \pi \cup \{a^\star\})$, and $a^\star$'s contribution counts on top. In bounded sets B^j with $s^j \leq |A^j \cap \pi| \leq u^j - 1$ (*i.e.*, where $a^\star$ is a substitute) we know that the contribution of B^j to $score_{tot}(\boldsymbol{B}, \pi)$ is s^j. This contribution is unchanged in $score_{tot}(\boldsymbol{B}, \pi \cup \{a^\star\})$. Thus, we can conclude that $score_{tot}(\boldsymbol{B}, \pi) \leq score_{tot}(\boldsymbol{B}, \pi \cup \{a^\star\})$. □

It turns out that also the other φ-based scoring functions $score_{min}$ and $score_{max}$ are not perfect from an axiomatic point as detailed in Table 1. The formal proofs are omitted due to space constraints.

Theorem 5. *The scoring function $score_{min}$ satisfies approval, incompatibility, dependency, and zero adequacy, as well as ballot-splitting monotonicity, but it fails substitution adequacy, ballot-size monotonicity, and score monotonicity.*

Theorem 6. *The scoring function $score_{max}$ satisfies substitution, incompatibility, dependency, and zero adequacy, as well as ballot-size monotonicity, score monotonicity, and ballot-splitting monotonicity, but it fails approval adequacy.*

Let us finally consider $score_{app}$. It fails several axioms, including substitution adequacy, as it completely ignores substitution effects.

Theorem 7. *The scoring function $score_{app}$ satisfies approval, incompatibility, and dependency adequacy, as well as ballot-size and ballot-splitting monotonicity. It fails substitution and zero adequacy, and score monotonicity.*

Proof. Approval adequacy follows directly from the definition of $score_{app}$. For ballot-size monotonicity, note that adding a bounded set never decreases the score. To prove the satisfaction of the other axioms, consider arbitrary ballot $\boldsymbol{B}$ and committee π.

Incompatibility Adequacy (✓) When we add bounded set $B^j = \langle A, 1, 1, 1\rangle$ to $\boldsymbol{B}$, the score cannot decrease as already stated above. However, it can increase only if there is some $a \in \pi$ with $a \in A^j$ and $1 = \ell^j \leq |A^j \cap \pi| \leq u^j = 1$, *i.e.*, if $|A \cap \pi| = 1$. If $|A \cap \pi| \neq 1$ it cannot increase. ◦

Dependency Adequacy (✓) When we add a bounded set indicating dependency $B^j = \langle A, |A|, |A|, |A|\rangle$ to $\boldsymbol{B}$, the score again cannot decrease. However, it can increase only if there is some $a \in \pi$ with $a \in A^j$ and $|A| = \ell^j \leq |A^j \cap \pi| \leq u^j = |A|$, *i.e.*, if $A \subseteq \pi$. If $A \nsubseteq \pi$ it cannot increase. ◦

Ballot-Splitting Monotonicity (✓) Let $B^j \in \boldsymbol{B}$ be a suitable set according to the definition of ballot-splitting monotonicity. Note that since $\ell^j \leq |A^j \cap \pi| \leq s^j$, each $a \in A^j \cap \pi$ contributes 1 to the score. By adding $\langle\{a\}, 1, 1, 1\rangle$ to the ballot, each a still contributes 1 to the score. However, for each $b \in \pi \setminus A$ there is no new bounded set concerning b, so they score just as before, too. All other alternatives in $\mathcal{A}$ score 0 just as before. Thus, the score is unchanged. ◦

Now we turn to the counterexample showing that substitution adequacy, zero adequacy, and score monotonicity are failed. Consider $\mathcal{A} = \{a_1, a_2, a_3\}$ and $\boldsymbol{B} = (\langle\{a_1, a_2\}, 1, 1, 2\rangle)$. Let $\pi = \{a_1\}$. Note that a_2 is a suitable substitute. Let $\pi' = \pi \cup \{a_2\}$. Since $score_{app}(\boldsymbol{B}, \pi) = 1 > score_{app}(\boldsymbol{B}, \pi') = 0$, substitution is failed. We also see that a score of 0 is possible even though all upper and lower bound are respected for the committee π'. Thus, zero adequacy is failed. Finally, since adding a_2 to π decreased the score though the bounds are respected, score monotonicity is also failed. □

3.3 Impossibility and Possibility Results

The fact that $score_{avg}$, $score_{min}$, $score_{max}$, and $score_{app}$ fail substitution but satisfy approval adequacy, and that $score_{tot}$ satisfies substitution but fails approval adequacy, is actually a hint at a bigger result: it is impossible to satisfy both approval adequacy and substitution adequacy at the same time.

Theorem 8. *No scoring function satisfies approval adequacy and substitution adequacy simultaneously.*

Proof. Let *score* be a scoring function that satisfies approval adequacy and substitution. Throughout the proof, we will consider an instance with three alternatives: $\mathcal{A} = \{a_1, a_2, a_3\}$. Let us first look at the following profile $\mathfrak{B}$ composed of the single voter's ballot:

$$\boldsymbol{B} = (\langle\{a_1, a_3\}, 1, 1, 2\rangle, \langle\{a_2, a_3\}, 1, 1, 2\rangle).$$

For $\pi_1 = \{a_3\}$ approval adequacy implies $score(\boldsymbol{B}, \pi_1) = 1$. Note that a_1 is a suitable substitute for π_1 as defined in Definition 5. Thus, for $\pi_1' = \{a_1, a_3\}$ must hold $score(\mathfrak{B}, \pi_1') = score(\mathfrak{B}, \pi_1) = 1$ in order for *score* to satisfy substitution. Interestingly, alternative a_2 is a suitable substitute for π_1'. Thus, for $\pi_1'' = \{a_1, a_2, a_3\}$ substitution entails that $score(\mathfrak{B}, \pi_1'') = score(\mathfrak{B}, \pi_1') = 1$. Consider now the committee $\pi_2 = \{a_1, a_2\}$. Approval adequacy on $\boldsymbol{B}$ and π_2 implies $score(\boldsymbol{B}, \pi_2) = 2$. Alternative a_3 is a suitable substitute here, thus, for $\pi_2' = \{a_1, a_2, a_3\}$ we have $score(\mathfrak{B}, \pi_2') = score(\mathfrak{B}, \pi_2) = 2$. Since $\pi_2' = \pi_1''$, the contradiction between is immediate. □

This impossibility is quite stringent as it prevents us from modeling what we had in mind in the first place. One way to circumvent it is to restrict the ballots. For instance, whenever bounded sets are not overlapping, *i.e.*, no alternative appears in more than one bounded set per ballot, then all φ-based scoring functions coincide and thus satisfy both substitution and approval adequacy (and also any axiom that is satisfied by at least one of them).

Theorem 9. *For every ballot $\boldsymbol{B}$ such that for any two bounded sets B^j and $B^{j'}$ in $\boldsymbol{B}$, we have $A^j \cap A^{j'} = \emptyset$, $score_{min}$, $score_{max}$, $score_{avg}$, and $score_{tot}$ coincide and thus all satisfy approval, substitution, incompatibility, dependency, and zero adequacy, as well as ballot-size, ballot-splitting, and score monotonicity.*

Another approach to "escape" the impossibility would be to weaken the axioms. We first investigate weak-substitution adequacy that requires the score not to improve when adding a substitute to the committee—instead of simply scoring the same.

Definition 11 (Weak-Substitution Adequacy)**.** *A scoring function score satisfies* weak-substitution adequacy *if for every ballot $\boldsymbol{B}$ and committee π for which there exists an alternative $a^\star \in \mathcal{A}\setminus\pi$ such that for all bounded sets $B^j \in \boldsymbol{B}_{|a^\star}$, it is the case that $s^j \leq |A^j \cap \pi| \leq u^j - 1$, we have $score(\boldsymbol{B}, \pi) \geq score(\boldsymbol{B}, \pi \cup \{a^\star\})$.*

It is clear that substitution adequacy implies weak substitution adequacy. Furthermore, it is easy to see that weak-substitution adequacy together with score monotonicity implies substitution adequacy. We can conclude that no scoring function can satisfy score monotonicity, weak-substitution, and approval adequacy at the same time. Finally, note that the counterexamples used to show that $score_{avg}$, $score_{min}$, and $score_{max}$ fail substitution adequacy also show that these scoring functions fail weak-substitution adequacy. We can however prove that $score_{app}$ satisfies it.

Table 1: Summary of our axiomatic analysis. Suits show the impossibility that all axioms with same suit are combined.

$score_{\mathbf{x}}$	*min*	*max*	*avg*	*tot*	*app*
App. Adeq. ♣ ♠	✓	✓	✓	✗	✓
Subst. Adeq. ♣	✗	✗	✗	✓	✗
Incomp. Adeq.	✓	✓	✓	✓	✓
Dep. Adeq.	✓	✓	✓	✓	✓
Zero Adeq.	✗	✓	✓	✓	✗
Weak-Subst. Adeq. ♠	✗	✗	✗	✓	✓
Weak-App. Adeq.	✓	✓	✓	✗	✓
Ballot-Size Mon.	✗	✓	✗	✓	✓
Ballot-Split. Mon.	✓	✓	✓	✓	✓
Score Mon. ♠	✗	✗	✗	✓	✗

Proposition 10. *The scoring function $score_{app}$ satisfies weak-substitution adequacy.*

Proof. Consider a substitute $a^\star \in \mathcal{A}\setminus\pi$ with $s_j \leq |A^j \cap \pi| < u^j$ for all $B^j \in \boldsymbol{B}_{|a^\star}$. If $a^\star$ is added to π, it holds that $|A^j \cap (\pi \cup \{a^\star\})| = |A^j \cap \pi| + 1$ for all $B^j \in \boldsymbol{B}_{|a^\star}$ and $|A^j \cap (\pi \cup \{a^\star\})| = |A^j \cap \pi|$ for all other B^j. Together with the fact that $a^\star$ is a substitute follow the following two facts. (1) If for an alternative $a \in \pi$ exists no set $B^j \in \boldsymbol{B}_{|a}$ s.t. $\ell^j \leq |A^j \cap \pi| \leq s^j$ there is also no set $B^j \in \boldsymbol{B}_{|a}$ s.t. $\ell^j \leq |A^j \cap (\pi \cup \{a^\star\})| \leq s^j$. (2) There is no set $B^j \in \boldsymbol{B}_{|a^\star}$ s.t. $\ell^j \leq |A^j \cap (\pi \cup \{a^\star\})| \leq s^j$. Thus, the score does not increase. □

We now consider weakening approval adequacy. The impossibility result is largely based on the fact that bounded sets are overlapping. One could thus weaken approval adequacy to require scoring functions to coincide with the usual approval score only when bounded sets are not overlapping.

Definition 12 (Weak-Approval Adequacy)**.** *A scoring function score satisfies* weak-approval adequacy *if for every ballot $\boldsymbol{B}$ and committee π the following two conditions hold:*

(1) $score(\boldsymbol{B}, \pi) \leq \left|\left(\bigcup_{B^j \in \boldsymbol{B}} A^j\right) \cap \pi\right|$;

(2) $score(\boldsymbol{B}, \pi) = \left|\left(\bigcup_{B^j \in \boldsymbol{B}} A^j\right) \cap \pi\right|$ *whenever it holds that both* $\ell^j \leq |A^j \cap \pi| \leq s^j$ *and* $A^j \cap A^{j'} = \emptyset$ *for all* $B^j, B^{j'} \in \boldsymbol{B}$.

Note that approval adequacy implies weak-approval adequacy.

Can we find a scoring function satisfying both weak-approval and substitution adequacy at the same time? Yes, we could use the following scoring function, that essentially forbids overlapping ballots:

$$score(\boldsymbol{B}, \pi) = \begin{cases} score_{tot}(\boldsymbol{B}, \pi) & \text{if } A^j \cap A^{j'} = \emptyset \ \ \forall B^j, B^{j'} \in \boldsymbol{B} \\ 0 & \text{otherwise.} \end{cases}$$

It is clear that this function satisfies substitution adequacy since $score_{tot}$ does (for the first case), and any constant scoring function does as well (for the second case). Weak-approval adequacy is also trivially satisfied. Indeed, the above scoring function scores non-0 only for profiles satisfying the second condition of weak-approval adequacy. This scoring function is quite artificial but proves possibility. We did not find an intuitively appealing scoring function that would satisfy both weak-approval and substitution adequacy.

4 EXPRESSIVENESS OF BOUNDED APPROVAL BALLOTS

We now want to study how expressive bounded approval ballots are. Because the way they are defined, we cannot discuss expressiveness of bounded approval ballots on their own, but need to consider them together with some scoring function.

In the following, we first study which kind of weak ordinal rankings over subsets of alternatives a bounded approval ballot—paired with a scoring function—can induce. Then, we show that the additional expressiveness can be crucial for the voters' satisfaction, when compared to standard approval ballots.

4.1 Limits to Expressiveness

For a given scoring function *score*, we study the limits of what a bounded approval ballot $\boldsymbol{B}$ can express. We work under the assumption that voters have ordinal preferences over committees and we measure expressiveness through the type of rankings over committees that can be induced by the scoring function, when ranking all committees based on their score for the ballot $\boldsymbol{B}$. Formally, every voter $i \in \mathcal{N}$ is equipped with a weak ranking over committees in C_k denoted by $\succeq_i$. We represent a weak ranking $\succeq$ over C_k as an ordered partition $\succeq = (C^1_\succeq, C^2_\succeq, \dots)$ of C_k where $C^1_\succeq$ contains the most preferred committees, and so on. The rank of a committee π in $\succeq$—denoted by $rank_\succeq(\pi)$—is the value $j \in \mathbb{N}$ such that $\pi \in C^j_\succeq$. For a given scoring function *score* and bounded approval ballot $\boldsymbol{B}$, let $\succeq^{score}_{\boldsymbol{B}}$ be the weak order over C_k such that for all $\pi, \pi' \in C_k$, $\pi \succeq^{score}_{\boldsymbol{B}} \pi'$ if and only if $score(\boldsymbol{B}, \pi) \geq score(\boldsymbol{B}, \pi')$. A scoring function can represent such a ranking $\succeq$ over C_k if there exists a bounded approval ballot $\boldsymbol{B}$ such that $\succeq^{score}_{\boldsymbol{B}}$ and $\succeq$ coincide. Everything is now set for us to delve into expressiveness. Our findings are summarized in Table 2. We start with arbitrary orders over C_k.

Proposition 11. *The scoring functions $score_{avg}$ and $score_{tot}$ can represent any arbitrary weak order $\succeq$ over C_k for any $k \geq 2$, while $score_{min}$, $score_{max}$, and $score_{app}$ cannot.*

Proof. Let $\succeq$ be an arbitrary ranking over C_k. To show that $score_{tot}$ can represent $\succeq$ we construct a ballot $\boldsymbol{B}$ as follows. For every $\pi \in C_k$, we add to $\boldsymbol{B}$ as many copies of $\langle \pi, k, k, k \rangle$ as $\binom{m}{k} - rank_\succeq(\pi)$. Since for $\pi, \pi' \in C_k$, it holds that $score_{tot}(\langle \pi, k, k, k \rangle, \pi')$ is k if $\pi = \pi'$ and zero if $\pi \neq \pi'$, the result then follows.

For $score_{avg}$, we extend the ballot described above, to bypass normalization, by enforcing that each alternative appears in an equal number of bounded sets. In particular, for all $\pi \in C_k$, we add sufficiently many copies of $\langle \pi, k-1, k-1, k-1 \rangle$ to $\boldsymbol{B}$, such that every $\pi \in C_k$ appears in exactly $\binom{m}{k}$ bounded sets in $\boldsymbol{B}$. This ensures that each alternative appears in exactly $\gamma = \binom{m}{k-1} \cdot \binom{m}{k}$ bounded sets. We thus have $\gamma \cdot score_{avg}(\boldsymbol{B}, \pi) = score_{tot}(\boldsymbol{B}, \pi)$ for all $\pi \in C_k$. The claim is thus derived from the above.

The claim for $score_{max}$ and $score_{app}$ follows from Proposition 12 (see below). For $score_{min}$, we can use a counting argument. Assume $\succeq$ is a strict ranking over C_k, *i.e.*, we have $|\succeq| = \binom{m}{k}$. We claim that for any fixed ballot $\boldsymbol{B}$, $score_{min}(\boldsymbol{B}, \pi)$ can take at most $\binom{k^2+k+1}{k}$ different values for different $\pi \in C_k$. This is because for a given bounded set $B = \langle A, \ell, s, u \rangle$, the size of the image of $\varphi(B, \pi)$, where π is the input, is at most $k^2 + 2$. Indeed, $\varphi(B, \pi)$ can be either 0, 1 or $s^j / |A^j \cap \pi|$ and that there are k^2 possible values for the latter (as if $s^j > k$, then $\varphi(B, \pi) \in \{0, 1\}$). For all k alternatives in π, $score_{min}$ takes the minimum of the relevant $\varphi(B, \pi)$ and then sum them up. The final score is then the sum of k (not necessarily distinct) values from a set of $k^2 + 2$ ones. Hence, the number of possible values for $score_{min}$ (and any $\pi \in C_k$) is bounded upwards by the number of multisets with cardinality k, taken from a set of size k^2+2. The latter is well known to be $\binom{k^2+k+1}{k}$, which is smaller than $|\succeq| = \binom{m}{k}$ as soon as $m > k^2 + k + 1$. This concludes the counting argument. □

Table 2: Expressiveness of the scoring functions.

$score_x$	*min*	*max*	*avg*	*tot*	*app*
Arbitrary	✗	✗	✓	✓	✗
Trichotomous	✓	✗	✓	✓	✗
Dichotomous	✓	✓	✓	✓	✓

To understand where the limit in expressiveness lies for $score_{min}$, $score_{max}$, and $score_{app}$ we focus on specific classes of orders over C_k. A weak order $\succeq$ is said to be *dichotomous* if $|\succeq| = 2$, and *trichotomous* if $|\succeq| = 3$. We show that $score_{min}$ can capture the former, while $score_{max}$ and $score_{app}$ can only capture the latter.

Proposition 12. *The scoring function $score_{min}$ can represent any trichotomous weak order $\succeq$ over C_k for any $k \geq 2$, while $score_{max}$ and $score_{app}$ cannot.*

Proof. To represent a trichotomous order $\succeq = (C^1_\succeq, C^2_\succeq, C^3_\succeq)$ over C_k with $score_{min}$, we construct a ballot $\boldsymbol{B}$ as follows. First, we add $\langle \{a\}, 1, 1, 1 \rangle$ for each $a \in \mathcal{A}$ to $\boldsymbol{B}$. For now, any committee $\pi \in C_k$ would have a score of k. We diminish the score of all committees $\pi \in C^2_\succeq$ by adding $\langle \pi, 1, k-1, k \rangle$ to $\boldsymbol{B}$. Note that this does not impact the score of any committee $\pi \in C^1_\succeq$. For any $\pi \in C_k$, it now holds that each $a \in \pi$ receives a score of one if and only if there is no $\langle \pi, 1, k-1, k \rangle \in \boldsymbol{B}$, and $k-1/k$ otherwise. Finally, for any $\pi \in C^3_\succeq$, we add $\langle \pi, 1, k-1, k-1 \rangle$ to $\boldsymbol{B}$, so that all these committees score 0. We thus have three levels of score: k for committees in $C^1_\succeq$, $k-1/k$ for committees in $C^2_\succeq$ and 0 for committees in $C^3_\succeq$.[2]

For $score_{max}$, consider $\mathcal{A} = \{a, b, c, d\}$, $k = 2$ and the order $\succeq$ such that $C^1_\succeq = \{\{a, b\}, \{c, d\}\}$, $C^2_\succeq = \{\{a, d\}, \{b, c\}\}$ and $C^3_\succeq = \{\{a, c\}, \{b, d\}\}$. Consider an arbitrary ballot $\boldsymbol{B}$. It is important to note that in this case, the score of a committee would always be a multiple of $1/2$ (because $\varphi(B, \pi) \in \{0, 1/2, 1\}$ for all B and π). If $score_{max}(\boldsymbol{B}, \{a, c\}) = score_{max}(\boldsymbol{B}, \{b, d\}) = 0$, then $\boldsymbol{B}$ may not contain a bounded set with a lower bound of 1. Hence, the remaining committees can all either receive a score of 0 or 2, which cannot lead to a trichotomous order. Next, it is easy to see, that no committee $\{x, y\} \in C_k$ can achieve a score of $1/2$. The only way x can receive a score of $1/2$, is with $\langle \{x, y\}, 1, 1, 2 \rangle$ and then y necessarily yields a score of at least $1/2$, too. Hence, in order to achieve said ordering, it must hold that $score_{max}(\boldsymbol{B}, \{a, c\}) = score_{max}(\boldsymbol{B}, \{b, d\}) = 1$ and $score_{max}(\boldsymbol{B}, \{a, d\}) = score_{max}(\boldsymbol{B}, \{b, c\}) = 3/2$. For the latter to

[2] Note that this would not work for $k = 1$ as in this case, $score_{min}(\boldsymbol{B}, \pi)$ can only take two values—0 or 1—for any $\boldsymbol{B}$ and π.

hold (*i.e.*, an alternative to yield a score of $1/2$), both $\langle\{a,d\},1,1,2\rangle$ and $\langle\{b,c\},1,1,2\rangle$ must be added to the ballot. This is a contradiction, because then $\{a,c\}$ yields a score of two.

For $score_{app}$ the situation is similar to $score_{max}$. To see that we are not able to represent trichotomous preferences, we can use the same counterexample, where only the first case applies. In particular, for trichotomous rankings and $k = 2$ we can always assume that the score for the least preferred committees must be zero, as $score_{app}(\boldsymbol{B}, \{x,y\}) \in \{0,1,2\}$ holds by design. □

Proposition 13. *The scoring functions $score_{max}$ and $score_{app}$ can represent any dichotomous weak order $\succeq$ over C_k.*

Proof. To represent any dichotomous order $\succeq$ with $score_{max}$ or $score_{app}$, we may add the bounded set $\langle\pi,k,k,k\rangle \in \boldsymbol{B}$ to the ballot, for each committee $\pi \in C^1_{\succeq}$. For both scoring functions, the score for a committee π would then be k if $\pi \in C^1_{\succeq}$ or 0 otherwise. □

4.2 Comparison to Approval Ballots

Bounded approval ballots are our proposal to provide voters more expressive ballots. It is clear that simple approval ballots cannot express more than bounded approval ballots, since the latter generalizes the former. But are approval ballots really so much weaker than bounded approval ballots? In the following we illustrate that approval ballots—even strategic ones—can lead to much worse results for the voters than bounded approval ballots.

Bounded approval ballots also impose a restriction on the preferences that can be expressed. This is why classical measures like *distortion* [17] cannot be used here to compare bounded and standard approval ballots. We will instead show with the following examples that *(i)* standard approval ballots are not sufficiently expressive, compared to bounded ones, especially in cases where communication between the voters is impossible, and *(ii)* the loss of expressiveness can largely impact on the voters' satisfaction.

Example 14. *(Pure Substitution)* Assume that for every voter i, their preferences are defined such that there exists a set of alternatives $A_i \subseteq \mathcal{A}$ for which i is unsatisfied whenever $\pi \cap A_i = \emptyset$ and fully satisfied as soon as $\pi \cap A_i \neq \emptyset$. Note that i's preferences can easily be expressed by a single bounded set $\langle A_i, 1, 1, |A_i|\rangle$. Now, if voter i were asked to submit a standard approval ballot, the only reasonable ballot to submit would be A_i.

Let the number of voters n be such that n is divisible by k, and let $\mathcal{A} = \{a_1,\dots,a_{k^2}\}$. Consider the profile $\mathfrak{B}$ of bounded approval ballots in which n/k voters submit $\langle\{a_1,\dots,a_k\},1,1,k\rangle$, n/k voters submit $\langle\{a_{k+1},\dots,a_{2k}\},1,1,k\rangle$, and so on. If standard approval ballots were used, the first group of voters would approve $\{a_1,\dots,a_k\}$, the second group $\{a_{k+1},\dots,a_{2k}\}$, and so on. Overall, all alternatives would be approved by the same number of voters. Thus, if we were to select a committee of size k that maximizes the social welfare[3], for a suitable tie-breaking rule[4], $\{a_1,\dots,a_k\}$ would be selected using standard approval ballots. This fully satisfies the first voter block, but no other voters. In the case of bounded approval ballots, we have $score(\mathfrak{B},\{a_1,\dots,a_k\}) = n/k$, but $score(\mathfrak{B},\{a_k,a_{2k},\dots,a_{k^2}\}) = n$ for any φ-based scoring function *score*. Thus, $\{a_k,a_{2k},\dots,a_{k^2}\}$ may satisfy *all* voters. △

The example above shows that already for preferences incorporating approval and substitution, it is possible that only a fraction of the voters that could be fully satisfied are satisfied in simple approval voting. This becomes even worse with incompatibilities.

Example 15. *(Pure Incompatibility)* Assume that for every voter i, their preferences are defined such that there exists a set of alternatives $A_i \subseteq \mathcal{A}$ for which i is unsatisfied whenever $|\pi \cap A_i| \neq 1$ and fully satisfied otherwise. Note that voter i's preferences can be expressed by a bounded set $\langle A_i,1,1,1\rangle$.

Let $n \geq 3$, $k = 2$, and $\mathcal{A} = \{a,b,c,d\}$. Assume that the first $n-1$ voters submit the ballot $\langle\{a,b\},1,1,1\rangle$, and the last voter submits $\langle\{c,d\},1,1,1\rangle$. It is clear that according to each of our scoring functions the committees maximizing the social welfare are $\{a,c\}$, $\{a,d\}$, $\{b,c\}$, or $\{b,d\}$. Each of them fully satisfies *all* voters. Under standard approval ballots it is reasonable to assume that the first $n-1$ voters would submit either $\{a\}$, $\{b\}$, or $\{a,b\}$, and the last one either $\{c\}$, $\{d\}$, or $\{c,d\}$. Then, unless the first $n-1$ voters all approve of only a or only b (which is unlikely if communication is impossible), the committee $\{a,b\}$ would maximize the social welfare. Note that it satisfies no voter at all. △

As voters cannot express incompatibilities in approval ballots, it is possible that all voters dislike the outcome, but there exists an outcome fully satisfying every voter. This massive difference comes solely from the bit of extra information in the bounded ballots.

5 CONCLUSIONS

We proposed bounded approval ballots as an extension to standard approval ballots. Bounded ballots are cognitively simple to use, and provide a reasonable surplus in expressiveness. Voters can easily express not only approval, but also substitution effects, incompatibilities, and dependencies between alternatives. We believe that these are the most common inter-alternative effects which voters want to express in multiwinner voting. Voters have indeed a high incentive to provide the extra information in bounded approval ballots, as it may greatly improve their satisfaction with the outcome.

We defined several scoring functions to evaluate bounded approval ballots. Our axiomatic study discovered that maintaining a behavior similar to that of standard approval ballots and simultaneously capturing substitution effects is generally impossible. This however seems to be a general problem that is not specific to bounded approval ballots. The most convenient way to circumvent the impossibility is to require bounded sets within a voter's ballots to be disjoint.

Of course, we want our ballot format to be used and tested in real elections. This will be the ultimate test to find out whether the ballot format is both simple and expressive enough to provide a real benefit for the voters. To allow interested research teams to make their own tests, we provide a prototype web-application in the following GitHub repository: github.com/claussmann/GoodVotes.

[3]For a profile of standard approval ballots $(A_i)_{i\in N}$, the social welfare for a committee π is defined as $\sum_{i\in N}|A_i \cap \pi|$. For a profile of bounded approval ballots $\mathfrak{B}$ with scoring function *score*, the social welfare of a committee π is defined as $\sum_{i\in N} score(\mathfrak{B},\pi)$.

[4]Note that we could also add an extra voter to the first group to make tie-breaking unnecessary. However, the result would then only hold asymptotically for large n.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers of AAMAS 2023 for their helpful comments. We also thank Jérôme Lang for having provided very useful feedback on a preliminary version of this paper. This work was supported in part by the project "Online Participation," funded by the NRW Ministry for Innovation, Science, and Research, by the "Deutsche Forschungsgemeinschaft (DFG)" under grants RO-1202/21-1 and BA-6270/1-1, and by the Dutch Research Council (NWO) under grant number 639.023.811. Current affiliation of the first author: Federal University of Applied Administrative Science, Brühl, Germany.

REFERENCES

[1] Haris Aziz, Markus Brill, Vincent Conitzer, Edith Elkind, Rupert Freeman, and Toby Walsh. 2017. Justified Representation in Approval-Based Committee Voting. *Social Choice and Welfare* 48, 2 (2017), 461–485.

[2] Haris Aziz and Nisarg Shah. 2020. Participatory Budgeting: Models and Approaches. In *Pathways between Social Science and Computational Social Science: Theories, Methods and Interpretations.*

[3] Nathanaël Barrot and Jérôme Lang. 2016. Conditional and Sequential Approval Voting on Combinatorial Domains. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI).* 88–94.

[4] Richard Booth, Yann Chevaleyre, Jérôme Lang, Jérôme Mengin, and Chattrakul Sombattheera. 2010. Learning Conditionally Lexicographic Preference Relations. In *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI).* 269–274.

[5] Craig Boutilier, Ronen I. Brafman, Holger H. Hoos, and David Poole. 1999. Reasoning With Conditional Ceteris Paribus Preference Statements. In *Proceedings of the 15th Annual Conference on Uncertainty in Artificial Intelligence (UAI).* 71–80.

[6] Craig Boutilier and Holger H. Hoos. 2001. Bidding Languages for Combinatorial Auctions. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI).* 1211–1217.

[7] Steven J. Brams and D. Marc Kilgour. 2015. Satisfaction Approval Voting. In *Mathematical and Computational Modeling: With Applications in Natural and Social Sciences, Engineering, and the Arts*, Roderick Melnik (Ed.). Wiley Online Library, 273–298.

[8] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel D. Procaccia (Eds.). 2016. *Handbook of Computational Social Choice.* Cambridge University Press.

[9] Peter Cramton, Yoav Shoham, and Richard Steinberg (Eds.). 2010. *Combinatorial Auctions.* MIT Press.

[10] Roy Fairstein, Reshef Meir, and Kobi Gal. 2021. Proportional Participatory Budgeting with Substitute Projects. *arXiv preprint arXiv:2106.05360* (2021).

[11] Piotr Faliszewski, Piotr Skowron, Arkadii Slinko, and Nimrod Talmon. 2017. Multiwinner Voting: A New Challenge for Social Choice Theory. In *Trends in Computational Social Choice*, Ulle Endriss (Ed.). AI Access Foundation, 27–47.

[12] Michael R. Garey and David S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman and Company.

[13] Holger H. Hoos and Craig Boutilier. 2000. Solving Combinatorial Auctions Using Stochastic Local Search. In *Proceedings of the 17th AAAI Conference on Artificial Intelligence (AAAI).* 22–29.

[14] Pallavi Jain, Krzysztof Sornat, and Nimrod Talmon. 2020. Participatory Budgeting with Project Interactions. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI).* 386–392.

[15] Martin Lackner and Piotr Skowron. 2022. *Multi-Winner Voting with Approval Preferences.* Springer-Verlag.

[16] Jean-François Laslier and M. Remzi Sanver (Eds.). 2010. *Handbook on Approval Voting.* Springer-Verlag.

[17] Ariel D. Procaccia and Jeffrey S. Rosenschein. 2006. The Distortion of Cardinal Preferences in Voting. In *Cooperative Information Agents X, 10th International Workshop, CIA.* 317–331.

[18] Tuomas Sandholm. 2002. Algorithm for Optimal Winner Determination in Combinatorial Auctions. *Artificial Intelligence* 135, 1-2 (2002), 1–54.

[19] Piotr Skowron and Piotr Faliszewski. 2017. Chamberlin–Courant Rule with Approval Ballots: Approximating the MaxCover Problem with Bounded Frequencies in FPT Time. *Journal of Artificial Intelligence Research* 60 (2017), 687–716.

[20] William Thomson. 2001. On the Axiomatic Method and its Recent Applications to Game Theory and Resource Allocation. *Social Choice and Welfare* 18, 2 (2001), 327–386.

CHAPTER 11

CONCLUSION

In this thesis, we studied the isolated research fields *multiwinner elections*, *participatory budgeting*, and *judgment aggregation*, as well as their relationships to one another. We identified five reoccurring research goals in Chapter 3, revolving around *axiomatic analysis* (Q1), *complexity of winner determination* (Q2), *complexity of manipulative interference* (Q3), *relationships between rules* (Q4), and *ballot design* (Q5). Reflecting back on our initial research goals, we contributed to partly answering our predefined questions a total of 23 times, spread over seven articles. Instead of repeating each contribution individually,[54] let us rearrange our individual findings into a more holistic overview. To do so, we first recap our results for each of our five research questions, one at a time. In particular, in the upcoming Section 11.1 we briefly summarize, how addressing each of those universal questions from different angles across multiple publications has led to partial answers from quite multifaceted perspectives. Subsequently, in Section 11.2 we complement our results, by showcasing how synergetic effects for selected results across our works may lead to further valuable insights and novel implications. Lastly, we suggest possible directions for future research in Section 11.3.

To improve readability throughout this chapter, we refer to Chapters instead of publications. Recall that we studied *"Irresolute Approval-based Budgeting"* [25] in Chapter 4, *"Complexity of Manipulative Interference in Participatory Budgeting"* [22] in Chapter 5, *"Time-Constrained Participatory Budgeting Under Uncertain Project Costs"* [24] in Chapter 6, *"Complexity of Sequential Rules in Judgment Aggregation"* [13, 14] in Chapter 7, *"Collective Combinatorial Optimisation as Judgment Aggregation"* [33, 34] in Chapter 8, *"Distortion in Attribute Approval Committee Elections"* [21] in Chapter 9, and *"Bounded Approval Ballots"* [23] in Chapter 10.

[54] Detailed discussions on each contribution can be found in Sections 4.4, 5.4, 6.4, 7.4, 8.4, 9.4, and 10.4,

11.1 Summary of Results

To recap, we addressed our initial isolated research questions as follows.

Question Q1: Axiomatic Analysis

We addressed Question Q1 in five publications, coming from five different angles. *(i)* In Chapter 4 we extended the axiomatic study for participatory budgeting to an irresolute context. *(ii)* In Chapter 7 we showed that the outcome for sequential judgment aggregation rules does not change, when issues supported by an underlying rule are permuted to the beginning of a processing order. *(iii)* In Chapters 6 and 10 we designed and studied use-case specific axiomatic properties to evaluate the behavior and adequacy of our respective models. *(iv)* In Chapter 8 we connected prominent judgment aggregation rules to its specializations inside the realms of multiwinner elections and participatory budgeting, allowing for an implicit transfer of axioms. Finally, *(v)* in Chapters 6 and 10 we were able to prove impossibility results, which were shown to be escapable through approximation, a weakening of axioms, or structural assumptions over the given preferences.

Question Q2: Complexity of Winner Determination

We addressed Question Q2 to varying degrees in all seven publications and tackled it mostly from two directions. In all chapters, we explicitly or implicitly answered questions regarding the computational complexity of determining an outcome of a voting rule. Although in some chapters, we only closed an open gap in related literature, in other chapters (i.e., Chapters 4, 6, 7, 9, and 10), we also introduced some of the investigated rules in the first place.

Question Q3: Complexity of Manipulative Interference

We addressed Question Q3 rather straightforwardly in three publications. To a preliminary degree, we studied questions revolving around the general existence of manipulative control actions in participatory budgeting in Chapter 4. We extended this study in Chapter 5, by complementing complexity results and providing a general upper bound proving scheme for related problems of manipulative interference. In Chapter 7, we studied the complexity associated with strategically altering the outcome of a sequential rule by controlling the underlying processing order.

Question Q4: Relationships Between Rules

We addressed Question Q4 in six publications, showcasing how relationships between rules appear in many forms, both within and across related research fields. Overall, our findings can be grouped into three different categories. *(i)* Within specific research fields, we showed in Chapter 4 that two separately studied rules in participatory budgeting coincide. Interestingly, we were able to demonstrate a similar result in Chapter 7, connecting two large classes of sequential and non-sequential judgment aggregation rules. In Chapter 10, we showed that different scoring rules for bounded approval ballots collapse to the same function, in case bounded sets do not overlap. *(ii)* Across related disciplines, our most significant results were obtained in Chapter 8. In particular, we showed that many rules, studied independently in different research fields, can be simulated by well-known judgment aggregation counterparts. In Chapter 4 we showed that three prominent rules for participatory budgeting based on maximization coincide with restricted domains of the budgeted maximum coverage problem [106]. Further, *(iii)* some rules introduced in this thesis are generalizations of prominent aggregation methods. The frameworks in Chapters 9 and 10 rely on preference formats, which extend common approval ballots. Respective rules modeled for those frameworks generalize voting rules for approval-based preferences in a similar way. In Chapter 6 we generalized the Method of Equal Shares to also work with uncertain project costs. Lastly, in Chapter 8 we studied a natural extension for standard judgment aggregation to study weighted, asymmetric rules. Along with a set of generalized rules, we also showed that applying the Chamberlin-Courant rule to a symmetric setting is not particularly useful.

Question Q5: Ballot Design

Clearly, we addressed Question Q5 in Chapters 9 and 10.[55] It is worth mentioning that underlying assumptions and suitable use-cases for the two separate frameworks differ strongly. For attribute approval elections we consider situations, where each alternative is equipped with distinct and quantifiable quality criteria. Furthermore, the voters' preferences are not explicitly given over the set of candidates, but more about the qualities desired to be present in a committee. In contrast, considering bounded approval ballots, the voters' preferences are more centered towards candidates. Yet, voters are able to specify more evolved preferences to express doubts about the composition of a committee.

11.2 Further Implications

To showcase how some seemingly unrelated results are interconnected, let us discuss what (further) implications we may derive from looking at the collection of articles contained in this thesis.

We showed in Chapter 7, that for the ranked agenda rule it is Σ_2^{P}-complete to decide, whether there is a tie-breaking order, such that a distinct issue appears in the outcome (even for a constant number of judges). In Chapter 8 we showed that the (asymmetric) ranked agenda rule, instantiated to a participatory budgeting setting, coincides with an efficiently computable greedy rule. Finally, in Chapter 5 we showed that problems

[55]Note that the framework for participatory budgeting with uncertain project costs, studied in Chapter 6, does not fall into this question. Yet, an interesting direction for further research would be to consider conditional approvals in this model (linked to the exact cost of a project).

of manipulative interference are generally in NP for efficiently computable rules (such as greedy rules), including a not explicitly studied variant of control by setting the tie-breaking order. We conclude, there is a complexity gap for this form of manipulative interference, when moving from participatory budgeting to judgment aggregation.

Of course, our results on manipulative interference in Chapter 5 interact with the axiomatic analysis in Chapter 4, by complementing computational aspects. Vice versa, by providing general upper bound schemes for rules and axiomatic properties yet to study, we provide a foundation for systematically studying problems of winner determination and manipulative interference.

For some isolated results, we can already derive implications across different frameworks by themselves, by changing the perspective into a more global view. Take for example our general upper bound proof schemes in Chapter 5. The scheme for winner determination does not rely on the overall combinatorial constraint in participatory budgeting. Hence, for voting rules that select an outcome based on maximizing an efficiently computable scoring function, this result translates to other collective combinatorial optimization problems, such as multiwinner elections or judgment aggregation. Connecting this observation to our generalized voting rules for attribute approval elections (see Chapter 9), we can easily derive Θ_2^p (or Δ_2^p for the weighted extension) as upper bound on the existential, irresolute winner determination problem (for all studied rules). Analogously, we may establish general upper bounds for our rules in Chapter 10 in the framework for bounded approval ballots.

As pointed out by Kagita, Pujari, Padmanabhan, Aziz, and Kumar [102], for attribute approval elections we can easily model (standard candidate) approval ballots by considering only one category, which holds exactly one unique attribute for each candidate. Hence, generalized rules defined in Chapter 9 coincide with their specializations, when restricted to common approval ballots. Although we did not study the attribute approval framework from a computational point of view, the latter allows us to transfer hardness results directly. As an example, Sonar, Dey, and Misra [160] showed that in multiwinner elections, it is Θ_2^p-hard to decide whether a given candidate is part of at least one outcome by the Chamberlin-Courant rule. This hardness inherits directly to one of our attribute approval rules. Similarly, coNP-hardness can be derived for the egalitarian approval voting rule, following our computational results in Chapter 8.

By a modular definition, this generality regarding upper bounds in Chapter 5 also holds for manipulative interference. The usage of an alteration function can easily be incorporated into similar settings. Further, if the number of valid alterations is a constant, finding a suitable manipulative action cannot be significantly harder than the determination of an outcome (by brute-forcing over a constant number of manipulative actions). Note that we may also restrict the search space for an alteration function artificially through parameterization. Therefore, relevant implications can be derived easily using parameterized complexity (see de Haan [93] for further reading).

Our axiomatic results for the hybrid greedy rules in Chapter 4 indicate that outputting the most appealing result across multiple rules may interfere with keeping desirable properties. Even if all underlying rules satisfy an axiomatic property stating *'if a condition is met, then an implication must hold'*, switching between rules might lead to edge cases, where only one satisfies the condition in the first place (while switching rules may prevent the implication to unfold).

In some articles, we already utilized prior results. For example, in Chapter 7, we first established an axiomatic result related to sequential judgment aggregation rules, namely that permuting supported issues to the beginning of a processing order does not change the outcome. Subsequently, this allowed us to identify a connection to non-sequential rules.

Recall that in Chapter 8, we used an instantiation result to inherit a matching lower bound of Δ_2^{P} on the complexity for winner determination from our work on participatory budgeting in Chapter 5 to (weighted, asymmetric) judgment aggregation. Further, the identified relationships between different rules can be transitive. By exploiting that the applied reduction relies on only one voter, the result immediately transferred from the weighted median rule to the weighted egalitarian median rule (as both rules coincide in case there is only one voter).

Similarly, in Chapter 4 we showed that two rules for participatory budgeting (based on maximization) coincide with restricted domains of the budgeted maximum coverage problem. In Chapter 8, we also showed that exactly those budgeting rules are instances of well-studied judgment aggregation rules. Hence, results on axiomatic properties and computational complexity may be unified for a more complete picture in all three fields.

11.3 Closing Remarks and Future Work

As closing remarks, we discuss promising directions for future work. To motivate our choice of suggested topics as outlook, let us begin with an analogy.

> *Imagine the vast field of theoretical computer science as an infinite dark space. Eventually, exceptional minds acquired universal results, which shine as tiny stars and illuminate their surroundings. At first, the emitted light was just strong enough to not navigate in the pitch black. Over the years, computer science has become one of the most active research fields. With every new result, the fuzzy boundary between what we can and cannot see is pushed a little further away. The key results of this thesis are no different. Building on a bright history of profound research in the area, the main problems investigated in this thesis were fortunately all in plain sight. Albeit little, the additional room illuminated by each new result is just enough to see several exciting follow-up questions.*

The publications contained in this thesis, are fairly multifaceted. Although any key result on its own may stimulate the investigation of specific follow-up problems, let us turn our attention to those directions whose potential insights might shine the brightest. In particular, we suggest to pursue the following two lines of research for future work.

First, one of the arguably most important task in computational social choice is, to understand the axiomatic behavior of aggregation rules (we discuss the importance of this task extensively in Section 3.1). Most significantly, (im)possibility [4] and characterization [66, 133] results act as landmarks in the assessment of what we can and cannot model. Apart from deriving respective results from scratch, in some cases we might be able to incorporate well-known facts from related research into a new context. By identifying relationships between (seemingly different) rules, research results may be extended or unified (see Section 3.4). Recall from Chapter 3, that multiwinner elections, participatory budgeting, and judgment aggregation can be perceived as frameworks for aggregating a list of fixed-size binary strings into a set of such binary strings (abiding some constraint). Following this simplicity, we strongly conjecture there are similar frameworks with surprisingly diverse use-cases throughout both, theoretical research and practical application. Hence, identifying those intersections for frameworks and rules can be a rewarding research direction with the potential to tie various fields of research closer together.

Second, a huge part of this work revolves around determining the computational complexity of problems relating to winner determination or manipulative interference (see Sections 3.2 and 3.3). Yet, we analyzed related questions mostly from the angle of worst-case (time) complexity. Although respective lower and upper bounds can be a crucial indicator of whether or not a problem is generally solvable in practice, we need to diversify this study for a more complete picture. As an example, let us briefly discuss prominent approaches from related literature for dealing with a computationally hard voting rule. Elkind, Lackner, and Peters [70] argue, that there are many elections where the voters' ballots abide a structure (e.g., a reasonable voter would only approve a continuous interval of candidates if they can be ordered by a left-right political spectrum). In turn, a restriction of the space of possible ballots might admit an efficiently computable algorithm. Interestingly, structured preferences may increase the complexity for manipulative interference in some cases, if the ballots must remain structured (e.g., after a valid bribery attempt [109]). A similar approach, focusing on the output instead of the input, is to further refine the set of feasible outcomes by restricting the constraint language. For example, de Haan [91] shows that several (generally hard) problems regarding winner determination in judgment aggregation become tractable if the constraint for feasibility is a Horn formula or in 2-CNF. Yet another way to escape hardness, is by limiting the remaining parameters (see de Haan [93] for an application of parameterized complexity to judgment aggregation). For example, Talmon and Faliszewski [166] mention, there is an implementation for one of their voting rules for participatory budgeting, that is only exponential in the number of projects (and thus efficiently computable if this number is a constant). Overall, studying computational aspects for related questions from more diverse directions may hold significant implications for practical applications.

BIBLIOGRAPHY

[1] R. Ahlswede and I. Wegener. *Search Problems*. John Wiley & Sons, Inc., 1987.

[2] E. Anshelevich, A. Filos-Ratsikas, N. Shah, and A. Voudouris. “Distortion in Social Choice Problems: The First 15 Years and Beyond”. In: *Proceedings of the 30th International Joint Conference on Artificial Intelligence*. Survey Track. ijcai.org, 2021, pp. 4294–4301.

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

[4] K. Arrow. *Social Choice and Individual Values*. John Wiley and Sons, 1951 (revised edition 1963).

[5] H. Aziz, M. Brill, V. Conitzer, E. Elkind, R. Freeman, and T. Walsh. “Justified Representation in Approval-based Committee Voting”. In: *Social Choice and Welfare* 48.2 (2017), pp. 461–485.

[6] H. Aziz, B. Lee, and N. Talmon. “Proportionally Representative Participatory Budgeting: Axioms and Algorithms”. In: *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems. 2018, pp. 23–31.

[7] H. Aziz and B. E. Lee. “Proportionally Representative Participatory Budgeting with Ordinal Preferences”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 6. AAAI Press, 2021, pp. 5110–5118.

[8] H. Aziz and N. Shah. “Participatory Budgeting: Models and Approaches”. In: *Pathways Between Social Science and Computational Social Science: Theories, Methods, and Interpretations*. Ed. by T. Rudas and P. Gábor. Springer, 2021, pp. 215–236.

[9] C. Badue, R. Guidolini, R. V. Carneiro, P. Azevedo, V. B. Cardoso, A. Forechi, L. Jesus, R. Berriel, T. M. Paixao, F. Mutz, et al. “Self-driving Cars: A Survey”. In: *Expert Systems with Applications* 165 (2021), pp. 113816.

[10] N. Barrot and J. Lang. “Conditional and Sequential Approval Voting on Combinatorial Domains”. In: *Proceedings of the 25th International Joint Conference on Artificial Intelligence*. ijcai.org, 2016, pp. 88–94.

[11] 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.

[12] 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.

[13] 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.

[14] D. Baumeister, L. Boes, and R. Weishaupt. “Complexity of Sequential Rules in Judgment Aggregation”. In: *The 8th International Workshop on Computational Social Choice*. Ed. by B. Zwicker and R. Meir. Haifa, Israel, 2021.

[15] D. Baumeister, T. Böhnlein, L. Rey, O. Schaudt, and A. Selker. “Minisum and Minimax Committee Election Rules for General Preference Types”. In: *Proceedings of the 22nd European Conference on Artificial Intelligence*. Short paper. IOS Press, 2016, pp. 1656–1657.

[16] D. Baumeister, G. Erdélyi, O. Erdélyi, and J. Rothe. “Computational Aspects of Manipulation and Control in Judgment Aggregation”. In: Springer-Verlag *L*ecture Notes in Artificial Intelligence #8176, 2013, pp. 71–85.

[17] D. Baumeister, G. Erdélyi, O. Erdélyi, and J. Rothe. “Control in Judgment Aggregation”. In: *Proceedings of the 6th European Starting AI Researcher Symposium*. IOS Press, 2012, pp. 23–34.

[18] D. Baumeister, G. Erdélyi, and J. Rothe. “How Hard is it to Bribe the Judges? A Study of the Complexity of Bribery in Judgment Aggregation”. In: Springer-Verlag *L*ecture Notes in Artificial Intelligence #6992, 2011, pp. 1–15.

[19] 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.

[20] 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. 197–317.

[21] D. Baumeister and L. Boes. “Distortion in Attribute Approval Committee Elections”. In: *Proceedings of the 22nd International Conference on Autonomous Agents and Multiagent Systems*. IFAAMAS, 2023, pp. 2649–2651.

[22] D. Baumeister, L. Boes, and J. Hillebrand. "Complexity of Manipulative Interference in Participatory Budgeting". In: *Proceedings of the 7th International Conference on Algorithmic Decision Theory*. Springer. 2021, pp. 424–439.

[23] D. Baumeister, L. Boes, C. Laußmann, and S. Rey. "Bounded Approval Ballots: Balancing Expressiveness and Simplicity for Multiwinner Elections". In: *Proceedings of the 22nd International Conference on Autonomous Agents and Multiagent Systems*. IFAAMAS, 2023, pp. 1400–1408.

[24] D. Baumeister, L. Boes, and C. Laußmann. "Time-Constrained Participatory Budgeting Under Uncertain Project Costs". In: *Proceedings of the 31st International Joint Conference on Artificial Intelligence*. ijcai.org, 2022, pp. 74–80.

[25] D. Baumeister, L. Boes, and T. Seeger. "Irresolute Approval-based Budgeting". In: *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems*. IFAAMAS, 2020, pp. 1774–1776.

[26] D. Baumeister and S. Dennisen. "Voter Dissatisfaction in Committee Elections." In: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems*. IFAAMAS, 2015, pp. 1707–1708.

[27] D. Baumeister, S. Dennisen, and L. Rey. "Winner Determination and Manipulation in Minisum and Minimax Committee Elections". In: *International Conference on Algorithmic Decision Theory*. Springer. 2015, pp. 469–485.

[28] G. Benadè, N. Itzhak, N. Shah, A. Procaccia, and Y. Gal. *Efficiency and Usability of Participatory Budgeting Methods*. Manuscript. 2018.

[29] G. Benadè, S. Nath, A. Procaccia, and N. Shah. "Preference Elicitation for Participatory Budgeting". In: *Proceedings of the 31st AAAI Conference on Artificial Intelligence*. AAAI Press, 2017, pp. 376–382.

[30] G. Benadè, S. Nath, A. D. Procaccia, and N. Shah. "Preference Elicitation for Participatory Budgeting". In: *Management Science* 67.5 (2021), pp. 2813–2827.

[31] D. Black. *The Theory of Committees and Elections*. Cambridge University Press, 1958.

[32] N. Boehmer, P. Faliszewski, L. Janeczko, and A. Kaczmarczyk. "Robustness of Participatory Budgeting Outcomes: Complexity and Experiments". In: *Proceedings of the 16th International Symposium on Algorithmic Game Theory*. Springer-Verlag, 2023, pp. 161–178.

[33] L. Boes, R. Colley, U. Grandi, J. Lang, and A. Novaro. "Collective Combinatorial Optimisation as Judgment Aggregation". In: *Proceedings of the 13th Multidisciplinary Workshop on Advances in Preference Handling*. Ed. by M. Ozturk, C. Labreuche, P. Viappiani, and S. Destercke. Vienna, Austria, 2022.

[34] L. Boes, R. Colley, U. Grandi, J. Lang, and A. Novaro. “Collective Combinatorial Optimisation as Judgment Aggregation”. In: *Annals of Mathematics and Artificial Intelligence* (2023).

[35] J.-C. de Borda. “Mémoire sur les Élections au Scrutin”. In: *Histoire de L’Académie Royale des Sciences, Paris* (1781). For english translation, see A. de Grazia [90].

[36] S. Botan, R. de Haan, M. Slavkovik, and Z. Terzopoulou. “Egalitarian Judgment Aggregation”. In: *Autonomous Agents and Multi-Agent Systems* 37.1 (2023), pp. 16:1–16:32.

[37] S. Botan, A. Novaro, and U. Endriss. “Group Manipulation in Judgment Aggregation”. In: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems*. IFAAMAS, 2016, pp. 411—419.

[38] C. Boutilier, I. Caragiannis, S. Haber, T. Lu, A. D. Procaccia, and O. Sheffet. “Optimal Social Choice Functions: A Utilitarian View”. In: *Artificial Intelligence* 227 (2015), pp. 190–213.

[39] C. Boutilier and H. H. Hoos. “Bidding Languages for Combinatorial Auctions”. In: *17th*. 2001, pp. 1211–1217.

[40] S. J. Brams and P. C. Fishburn. “Approval Voting”. In: *American Political Science Review* 72.3 (1978), pp. 831–847.

[41] S. J. Brams, D. M. Kilgour, and M. R. Sanver. “A Minimax Procedure for Electing Committees”. In: *Public Choice* 132.3-4 (2007), pp. 401–420.

[42] S. J. Brams and M. R. Sanver. “Voting Systems that Combine Approval and Preference”. In: *The mathematics of Preference, Choice and Order: Essays in Honor of Peter C. Fishburn* (2009), pp. 215–237.

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

[44] R. Bredereck, P. Faliszewski, A. Igarashi, M. Lackner, and P. Skowron. “Multiwinner Elections with Diversity Constraints”. In: *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*. AAAI Press, 2018, pp. 933–940.

[45] M. Brill, S. Forster, M. Lackner, J. Maly, and J. Peters. “Proportionality in Approval-Based Participatory Budgeting”. In: *Proceedings of the 37th AAAI Conference on Artificial Intelligence*. AAAI Press, 2023, pp. 5524–5531.

[46] S. R. Buss and L. Hay. “On Truth-Table Reducibility to SAT”. In: *Information and Computation* 91.1 (1991), pp. 86–102.

[47] J. Cai, T. Gundermann, J. Hartmanis, L. Hemachandra, V. Sewelson, K. Wagner, and G. Wechsung. “The Boolean Hierarchy I: Structural Properties”. In: 17.6 (1988), pp. 1232–1252.

[48] J. Cai, T. Gundermann, J. Hartmanis, L. Hemachandra, V. Sewelson, K. Wagner, and G. Wechsung. "The Boolean Hierarchy II: Applications". In: 18.1 (1989), pp. 95–111.

[49] O. Cailloux and U. Endriss. "Arguing About Voting Rules". In: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems*. IFAAMAS, 2016, pp. 287–295.

[50] J. R. Chamberlin and P. N. Courant. "Representative Deliberations and Representative Decisions: Proportional Representation and the Borda Rule". In: *American Political Science Review* 77.3 (1983), pp. 718–733.

[51] J. Chingoma, U. Endriss, and R. de Haan. "Simulating Multiwinner Voting Rules in Judgment Aggregation". In: *Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems*. IFAAMAS, 2022, pp. 263–271.

[52] N. Chomsky. "On Certain Formal Properties of Grammars". In: *Information and Control* 2.2 (1959), pp. 137–167.

[53] A. Church. "An Unsolvable Problem of Elementary Number Theory". In: *American Journal of Mathematics* 58 (1936), pp. 345–363.

[54] R. Colley. "Expressive and Rational Delegations in Voting". PhD thesis. Université Toulouse Capitole, 2023.

[55] M. J. A. N. de Caritat (Marquis de Condorcet). "Essai sur l'Application de L'Analyse à la Probabilité des Décisions Rendues à la Pluralité des Voix". In: An english translation appears in I. McLean and A. Urken [129], pp. 91–112. 1785.

[56] V. Conitzer, W. Sinnott-Armstrong, J. S. Borg, Y. Deng, and M. Kramer. "Moral Decision Making Frameworks for Artificial Intelligence". In: *Proceedings of the 31st AAAI Conference on Artificial Intelligence*. AAAI Press, 2017, pp. 4831–4835.

[57] 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.

[58] 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.

[59] S. B. Cooper. *Computability Theory*. CRC Press, 2003.

[60] F.-A. Croitoru, V. Hondru, R. T. Ionescu, and M. Shah. "Diffusion Models in Vision: A Survey". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).

[61] N. Dias. *Hope for Democracy: 30 Years of Participatory Budgeting Worldwide*. 2018.

[62] N. Dias, S. Enríquez, and S. Júlio. “The Participatory Budgeting World Atlas”. In: *Epopeia and Oficina* (2019).

[63] F. Dietrich and C. List. “Arrow’s Theorem in Judgment Aggregation”. In: *Social Choice and Welfare* 29.1 (2007), pp. 19–33.

[64] F. Dietrich and C. List. “Judgment Aggregation by Quota Rules: Majority Voting Generalized”. In: *Journal of Theoretical Politics* 19.4 (2007), pp. 391–424.

[65] F. Dietrich. “Scoring Rules for Judgment Aggregation”. In: *Social Choice and Welfare* 42.4 (2014), pp. 873–911.

[66] F. Dietrich and C. List. “Judgment Aggregation by Quota Rules: Majority Voting Generalized”. In: *Journal of Theoretical Politics* 19.4 (2007), pp. 391–424.

[67] C. Dodgson. “A Method of Taking Votes on more than two Issues”. In: *The theory of committees and elections* (1876). Reprinted in [31] and [129].

[68] E. Dokow and R. Holzman. “Aggregation of binary evaluations”. In: *Journal of Economic Theory* 145.2 (2010), pp. 495–511.

[69] E. Dokow and R. Holzman. “Aggregation of Binary Evaluations for Truth-functional Agendas”. In: *Social Choice and Welfare* 32.2 (2009), pp. 221–241.

[70] E. Elkind, M. Lackner, and D. Peters. “Structured Preferences”. In: *Trends in Computational Social Choice*. Ed. by U. Endriss. AI Access, 2017. Chap. 10, pp. 187–207.

[71] 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.

[72] 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, 2016, pp. 176–185.

[73] U. Endriss, U. Grandi, and D. Porello. “Complexity of Judgment Aggregation”. In: *Journal of Artificial Intelligence Research* 45 (2012), pp. 481–514.

[74] 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.

[75] 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.

[76] U. Endriss. “Judgment Aggregation with Rationality and Feasibility Constraints”. In: *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*. IFAAMAS, 2018, pp. 946–954.

[77] P. Everaere, S. Konieczny, and P. Marquis. “Counting Votes for Aggregating Judgments”. In: *Proceedings of the 13th International Conference on Autonomous Agents and Multi-Agent Systems*. IFAAMAS, 2014, pp. 1177–1184.

[78] R. Fairstein, G. Benadè, and K. Gal. “Participatory Budgeting Design for the Real World”. In: *Proceedings of the 37th AAAI Conference on Artificial Intelligence*. AAAI Press, 2023, pp. 5633–5640.

[79] P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra. “How Hard is Bribery in Elections?” In: *Journal of Artificial Intelligence Research* 35 (2009), pp. 485–532.

[80] P. Faliszewski, E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. “Llull and Copeland Voting Computationally Resist Bribery and Constructive Control”. In: *Journal of Artificial Intelligence Research* 35 (2009), pp. 275–341.

[81] P. Faliszewski, P. Skowron, A. Slinko, and N. Talmon. “Multiwinner Voting: A New Challenge for Social Choice Theory”. In: *Trends in Computational Social Choice*. Ed. by U. Endriss. AI Access, 2017. Chap. 2, pp. 27–47.

[82] A. Fiat and G. J. Woeginger. *Online Algorithms: The State of the Art*. Vol. 1442. Springer, 1998.

[83] T. Fluschnik, P. Skowron, M. Triphaus, and K. Wilker. “Fair Knapsack”. In: *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*. Vol. 33. AAAI Press, 2019, pp. 1941–1948.

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

[85] A. Gibbard. “Manipulation of Voting Schemes: A General Result”. In: *Econometrica* 41.4 (1973), pp. 587–601.

[86] K. Gkiotsalitis. *Public Transport Optimization*. Springer Nature, 2023.

[87] J. Gomez, D. R. Insua, and C. Alfaro. “A Participatory Budget Model Under Uncertainty”. In: *European Journal of Operational Research* 249.1 (2016), pp. 351–358.

[88] U. Grandi and U. Endriss. “Lifting Integrity Constraints in Binary Aggregation”. In: *Artificial Intelligence* 199-200 (2013), pp. 45–66.

[89] U. Grandi. “Binary Aggregation with Integrity Constraints”. PhD thesis. 2012.

[90] A. de Grazia. “Mathematical Derivation of an Election System”. In: *Isis* 44.1-2 (1953), pp. 42–51.

[91] 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.

[92] 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.

[93] R. de Haan. *Parameterized Complexity in the Polynomial Hierarchy*. Springer, 2019.

[94] G. Hägele and F. Pukelsheim. “The Electoral Writings of Ramon Llull”. In: *Studia Lulliana* 41.97 (2001), pp. 3–38.

[95] M. A. Harrison. *Introduction to Formal Language Theory*. Addison-Wesley Longman Publishing Co., Inc., 1978.

[96] J. Hartmanis and R. Stearns. “On the Computational Complexity of Algorithms”. In: *Transactions of the American Mathematical Society* 117 (1965), pp. 285–306.

[97] L. Hemachandra. “The Strong Exponential Hierarchy Collapses”. In: ACM Press, 1987, pp. 110–122.

[98] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. “Exact Analysis of Dodgson Elections: Lewis Carroll’s 1876 Voting System is Complete for Parallel Access to NP”. In: 44.6 (1997), pp. 806–825.

[99] D Hilbert and W Ackerman. *Grundzüge der Theoretischen Logik*. Springer, 1928.

[100] H. H. Hoos and C. Boutilier. “Solving Combinatorial Auctions Using Stochastic Local Search”. In: *17th*. AAAI Press, 2000, pp. 22–29.

[101] J. Kadin. “The Polynomial Time Hierarchy Collapses if the Boolean Hierarchy Collapses”. In: *SIAM Journal on Computing* 17.6 (1988), pp. 1263–1282.

[102] V. Kagita, A. Pujari, V. Padmanabhan, H. Aziz, and V. Kumar. “Committee Selection using Attribute Approvals”. In: *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*. IFAAMAS, 2021, pp. 683–691.

[103] 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.

[104] H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack Problems*. Springer, 2004.

[105] J. Kemeny. “Mathematics Without Numbers”. In: *Dædalus* 88.4 (1959), pp. 577–591.

[106] S. Khuller, A. Moss, and J. S. Naor. “The Budgeted Maximum Coverage Problem”. In: *Information processing letters* 70.1 (1999), pp. 39–45.

[107] L. Kornhauser and L. Sager. “Unpacking the Court”. In: *Yale Law Journal* 96.1 (1986), pp. 82–117.

[108] M. Krentel. “The Complexity of Optimization Problems”. In: *Journal of Computer and System Sciences* 36 (1988), pp. 490–509.

[109] B. Kusek, R. Bredereck, P. Faliszewski, A. Kaczmarczyk, and D. Knop. “Bribery Can Get Harder in Structured Multiwinner Approval Election”. In: *Proceedings of the 22nd International Conference on Autonomous Agents and Multiagent Systems*. IFAAMAS, 2023, pp. 1725–1733.

[110] M. Lackner, J. Maly, and S. Rey. “Fairness in Long-Term Participatory Budgeting”. In: *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*. 2021, pp. 1566–1568.

[111] M. Lackner and P. Skowron. *Multi-winner Voting with Approval Preferences*. Springer Nature, 2023.

[112] R. Ladner, N. Lynch, and A. Selman. “A Comparison of Polynomial Time Reducibilities”. In: *Theoretical Computer Science* 1.2 (1975), pp. 103–124.

[113] R. E. Ladner and N. A. Lynch. “Relativization of Questions About Log Space Computability”. In: *Mathematical Systems Theory* 10 (1976), pp. 19–32.

[114] E. Landau. *Handbuch der Lehre von der Verteilung der Primzahlen*. Vol. 1. BG Teubner, 1909.

[115] J. Lang, G. Pigozzi, M. Slavkovik, and L. Van der Torre. “Judgment Aggregation Rules Based on Minimization”. In: *Proceedings of the 13th Conference on Theoretical Aspects of Rationality and Knowledge*. 2011, pp. 238–246.

[116] J. Lang, G. Pigozzi, M. Slavkovik, L. van der Torre, and S. Vesic. “A Partial Taxonomy of Judgment Aggregation Rules and their Properties”. In: *Social Choice and Welfare* 48.2 (2017), pp. 327–356.

[117] J. Lang and P. Skowron. “Multi-Attribute Proportional Representation”. In: *Artificial Intelligence* 263 (2018), pp. 74–106.

[118] J. Lang and M. Slavkovik. “How Hard is it to Compute Majority-Preserving Judgment Aggregation Rules?” In: *Proceedings of the Twenty-first European Conference on Artificial Intelligence*. IOS Press. 2014, pp. 501–506.

[119] J. Lang and M. Slavkovik. “Judgment Aggregation Rules and Voting Rules”. In: *Proceedings of the 3rd International Conference on Algorithmic Decision Theory*. Springer. 2013, pp. 230–243.

[120] J.-F. Laslier and M. R. Sanver. *Handbook on Approval Voting*. Springer Science & Business Media, 2010.

[121] C. Laußmann. “COMSOC Methods in Real-World Applications”. PhD thesis. 2023.

[122] C. Laußmann and P. Nüsken. *GoodVoteX*. `www.github.com/claussmann/GoodVoteX`. 2023.

[123] L. Levin. “Universal Sorting Problems”. In: *Problemy Peredaci Informacii* 9 (1973). For english translation, see B. Trakhtenbrot [168], pp. 115–116.

[124] C. List and P. Pettit. “Aggregating Sets of Judgments: An Impossibility Result”. In: *Economics and Philosophy* 18.1 (2002), pp. 89–110.

[125] C. List. “A Model of Path-Dependence in Decisions over Multiple Propositions”. In: *American Political Science Review* 98.3 (2004), pp. 495–513.

[126] C. List and C. Puppe. “Judgment Aggregation: A Survey”. In: *The Handbook of Rational and Social Choice*. Oxford University Press, 2009, pp. 457—482.

[127] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig. “Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing”. In: *ACM Computing Surveys* 55.9 (2023), pp. 1–35.

[128] T. Lukasiewicz and E. Malizia. “A Novel Characterization of the Complexity Class Θ_k^P Based on Counting and Comparison”. In: *Theoretical Computer Science* 694 (2017), pp. 21–33.

[129] I. McLean and A. Urken. *Classics of Social Choice*. Ann Arbor, Michigan: University of Michigan Press, 1995.

[130] A. R. Meyer and L. J. Stockmeyer. “The Equivalence Problem for Regular Expressions with Squaring Requires Exponential Space”. In: *Proceedings of the 13th IEEE Symposium on Switching and Automata Theory*. Vol. 72. 1972, pp. 125–129.

[131] M. K. Miller and D. Osherson. “Methods for Distance-Based Judgment Aggregation”. In: *Social Choice and Welfare* 32.4 (2009), pp. 575.

[132] K. Nehring and M. Pivato. “Majority Rule in the Absence of a Majority”. In: *Journal of Economic Theory* 183 (2019), pp. 213–257.

[133] K. Nehring and M. Pivato. “The Median Rule in Judgement Aggregation”. In: *Economic Theory* (2021).

[134] K. Nehring, M. Pivato, and C. Puppe. “The Condorcet Set: Majority Voting over Interconnected Propositions”. In: *Journal of Economic Theory* 151 (2014), pp. 268–303.

[135] J. Ober. *The Athenian Revolution: Essays on Ancient Greek Democracy and Political Theory*. Princeton University Press, 1999.

[136] C. Papadimitriou. *Computational Complexity*. 2nd ed. Addison-Wesley, 1995.

[137] C. Papadimitriou and D. Wolfe. "The Complexity of Facets Resolved". In: *Journal of Computer and System Sciences* 37.1 (1988), pp. 2–13.

[138] 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.

[139] C. H. Papadimitriou. "On the Complexity of Unique Solutions". In: *Journal of the ACM* 31.2 (1984), pp. 392–400.

[140] C. H. Papadimitriou and M. Yannakakis. "The Complexity of Facets (and some Facets of Complexity)". In: *Proceedings of the 14th Annual ACM Symposium on Theory of Computing*. 1982, pp. 255–260.

[141] B. Peleg and S. Zamir. "Judgments Aggregation by a Sequential Majority Procedure". In: *Mathematical Social Sciences* 95 (2018), pp. 37–46.

[142] D. Peters, G. Pierczyński, and P. Skowron. "Proportional Participatory Budgeting with Additive Utilities". In: *Advances in Neural Information Processing Systems* 34 (2021). Updated version arXiv:2008.13276, pp. 12726–12737.

[143] D. Peters, A. D. Procaccia, A. Psomas, and Z. Zhou. "Explainable Voting". In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 1525–1534.

[144] D. Peters and P. Skowron. "Proportionality and the Limits of Welfarism". In: *Proceedings of the 21st ACM Conference on Economics and Computation*. Extended version arXiv:1911.11747. ACM, 2020, pp. 793–794.

[145] G. Pigozzi. "Belief Merging and the Discursive Dilemma: An Argument-based Account to Paradoxes of Judgment Aggregation". In: *Synthese* 152 (2006), pp. 285–298.

[146] D. Porello and U. Endriss. "Ontology Merging as Social Choice: Judgment Aggregation under the Open World Assumption". In: *Journal of Logic and Computation* 24.6 (2014), pp. 1229–1249.

[147] A. Procaccia and J. Rosenschein. "The Distortion of Cardinal Preferences in Voting". In: *Proceedings of the 10th International Workshop on Cooperative Information Agents*. Springer-Verlag *Lecture Notes in Artificial Intelligence #4149*, 2006, pp. 317–331.

[148] S. Rey, U. Endriss, and R. de Haan. "A General Framework for Participatory Budgeting with Additional Constraints". In: *Social Choice and Welfare* (2023), pp. 1–37.

[149] S. Rey, U. Endriss, and R. de Haan. "Designing Participatory Budgeting Mechanisms Grounded in Judgment Aggregation". In: *Proceedings of the 17th International Conference on Principles of Knowledge Representation and Reasoning*. ijcai.org, 2020, pp. 692–702.

[150] S. Rey and J. Maly. “The (Computational) Social Choice Take on Indivisible Participatory Budgeting”. In: *arXiv preprint arXiv:2303.00621* (2023).

[151] J. Rothe. *Complexity Theory and Cryptology. An Introduction to Cryptocomplexity*. EATCS Texts in Theoretical Computer Science. Springer-Verlag, 2005.

[152] 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.

[153] T. Sandholm. “Algorithm for Optimal Winner Determination in Combinatorial Auctions”. In: *Artificial Intelligence* 135.1-2 (2002), pp. 1–54.

[154] 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.

[155] M. Schaefer and C. Umans. “Completeness in the Polynomial-Time Hierarchy: A Compendium”. In: *SIGACT news* 33.3 (2002), pp. 32–49.

[156] A. Shah. *Participatory Budgeting*. World Bank Publications, 2007.

[157] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis. “Mastering the Game of Go Without Human Knowledge”. In: *Nature* 550.7676 (2017), pp. 354–359.

[158] Y. Sintomer, C. Herzberg, and A. Röcke. “Participatory Budgeting in Europe: Potentials and Challenges”. In: *International Journal of Urban and Regional Research* 32.1 (2008), pp. 164–178.

[159] P. Skowron and P. Faliszewski. “Chamberlin–Courant Rule with Approval Ballots: Approximating the MaxCover Problem with Bounded Frequencies in FPT Time”. In: *Journal of Artificial Intelligence Research* 60 (2017), pp. 687–716.

[160] C. Sonar, P. Dey, and N. Misra. “On the Complexity of Winner Verification and Candidate Winner for Multiwinner Voting Rules”. In: *Proceedings of the 29th International Joint Conference on Artificial Intelligence*. ijcai.org, 2020.

[161] B. de Sousa Santos. “Participatory Budgeting in Porto Alegre: Toward a Redistributive Democracy”. In: *Politics & Society* 26.4 (1998), pp. 461–510.

[162] G. Sreedurga, M. R. Bhardwaj, and Y. Narahari. “Maxmin Participatory Budgeting”. In: *Proceedings of the 31st International Joint Conference on Artificial Intelligence*. ijcai.org, 2022, pp. 489–495.

[163] R. E. Stearns, J. Hartmanis, and P. M. Lewis. “Hierarchies of Memory Limited Computations”. In: *Proceedings of the 6th Annual Symposium on Switching Circuit Theory and Logical Design*. IEEE. 1965, pp. 179–190.

[164] L. Stockmeyer. “The Polynomial-Time Hierarchy”. In: *Theoretical Computer Science* 3.1 (1977), pp. 1–22.

[165] N. Talmon and P. Faliszewski. *A Framework for Approval-based Budgeting Methods*. Tech. rep. 2018. arXiv: `1809.04382`.

[166] N. Talmon and P. Faliszewski. “A Framework for Approval-based Budgeting Methods”. In: *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*. Vol. 33. AAAI Press, 2019, pp. 2181–2188.

[167] T. N. Thiele. “Om Flerfoldsvalg”. In: *Oversigt over det Kongelige Danske Videnskabernes Selskabs Forhandlinger* 1895 (1895), pp. 415–441.

[168] 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.

[169] 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.

[170] A. M. Turing. “Systems of Logic Based on Ordinals”. In: *Proceedings of the London Mathematical Society, Series 2* 45 (1939), pp. 161–228.

[171] L. G. Valiant. “The Complexity of Computing the Permanent”. In: *Theoretical Computer Science* 8.2 (1979), pp. 189–201.

[172] K. W. Wagner. “More Complicated Questions About Maxima and Minima, and some Closures of NP”. In: *Theoretical Computer Science* 51.1-2 (1987), pp. 53–80.

[173] B. Wampler, S. McNulty, and M. Touchton. *Participatory Budgeting in Global Perspective*. Oxford University Press, 2021.

[174] R. Weishaupt. “Deciding the Uncertain: Axiomatic Aspects of Fair Division and Computational Complexity Studies in Computational Social Choice and Graph Theory”. PhD thesis. 2022.

[175] C. Wrathall. “Complete Sets and the Polynomial-Time Hierarchy”. In: *Theoretical Computer Science* 3.1 (1976), pp. 23–33.

[176] D. Zhao, Y. Dai, and Z. Zhang. “Computational Intelligence in Urban Traffic Signal Control: A Survey”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 42.4 (2011), pp. 485–494.

[177] W. Zwicker. “Introduction to the Theory of 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. 2, pp. 23–56.

OMITTED PROOFS AND SUPPLEMENTARY RESULTS

In this appendix, we complement our works that have only been published as an extended abstract by providing formal proofs for our claims, as well as investigating some additional results. In particular, this relates to our work on *"Irresolute Approval-based Budgeting"* [25] (see Chapter 4) and our work on *"Distortion in Attribute Approval Committee Elections"* [21] (see Chapter 9).

A.1 Irresolute Approval-based Budgeting

In this section, we supplement our publication [25], by providing missing proofs (that were omitted due to space constraints). As an intermediate result, we formally link the *budgeted maximum coverage problem* by Khuller, Moss, and Naor [106] to our rules based on maximization to adopt an approximation result.

Approximation Results

Proposition 3.1 (in [25]). *$\mathcal{R}^g_{|B_v|}$, $\mathcal{R}^g_{\mathbb{1}_{|B_v|>0}}$, $\mathcal{R}^g_{c(B_v)}$, $\mathcal{R}^p_{|B_v|}$, $\mathcal{R}^p_{\mathbb{1}_{|B_v|>0}}$, and $\mathcal{R}^p_{c(B_v)}$ do not have a constant approximation factor.*

Proof. Let us begin with the rules $\mathcal{R}^g_{|B_v|}$ and $\mathcal{R}^g_{\mathbb{1}_{|B_v|>0}}$. Consider a budgeting scenario $E_1 = (A, V, c, \ell)$ with $A = \{a_0, a_1, \ldots, a_\ell\}$ and $V = \{v, v', v_1, v_2, \ldots, v_\ell\}$. The cost function is given by $c(a_0) = \ell$ and $c(a_i) = 1$ for every $i \neq 0$, and the voters' approval ballots by $A_v = A_{v'} = \{a_0\}$ and $A_{v_i} = \{a_i\}$ for all $i \in [\ell]$. Then for max rules we obtain $\mathcal{R}^m_{|B_v|}(E_1) = \mathcal{R}^m_{\mathbb{1}_{|B_v|>0}}(E_1) = \{\{a_1, \ldots, a_\ell\}\}$ with a total satisfaction of ℓ, while given greedy rules select the most expensive item in the first iteration, resulting in $\mathcal{R}^g_{|B_v|}(E_1) = \mathcal{R}^g_{\mathbb{1}_{|B_v|>0}}(E_1) = \{\{a_0\}\}$ with a total satisfaction of 2. Choosing ℓ arbitrarily large reveals, that the considered greedy rules cannot achieve a constant approximation factor.

We continue with $\mathcal{R}^{g}_{c(B_v)}$, by modifying the above budgeting scenario E_1 as follows. By replacing each voter $v_i \in V \setminus \{v, v'\}$ with ℓ identical clones, we obtain a budgeting scenario E_2, where the max rule $\mathcal{R}^{m}_{c(B_v)}$ results in a winning budget $\mathcal{R}^{m}_{c(B_v)}(E_2) = \{\{a_1, \ldots, a_\ell\}\}$ with a total satisfaction of ℓ^2, while the greedy rule $\mathcal{R}^{g}_{c(B_v)}$ selects the most expensive item in the first iteration, resulting in $\mathcal{R}^{g}_{c(B_v)}(E_2) = \{\{a_0\}\}$ with a total satisfaction of 2ℓ. Again, ℓ can be chosen arbitrarily large.

Next, we show the claim for the remaining rules $\mathcal{R}^{p}_{|B_v|}$ and $\mathcal{R}^{p}_{\mathbb{1}_{|B_v|>0}}$. Let $E_3 = (A, V, c, \ell)$ be a budgeting scenario with $A = \{a_1, a_2\}$, $V = \{v_0, \ldots, v_{\ell+1}\}$, $c(a_1) = 1$ and $c(a_2) = \ell$, and $A_{v_i} = \{a_1\}$ if $i \leq 1$ and $A_{v_i} = \{a_2\}$ otherwise. Then for max rules we get $\mathcal{R}^{m}_{|B_v|}(E_3) = \mathcal{R}^{m}_{\mathbb{1}_{|B_v|>0}}(E_3) = \{\{a_2\}\}$ with a total satisfaction of ℓ, while the proportional greedy rules select item a_1, since it has the better satisfaction-to-cost ratio, resulting in $\mathcal{R}^{p}_{|B_v|}(E_3) = \mathcal{R}^{p}_{\mathbb{1}_{|B_v|>0}}(E_3) = \{\{a_1\}\}$ with a total satisfaction of 2.

Finally, for $\mathcal{R}^{p}_{c(B_v)}$ we modify the budgeting scenario E_3, such that a_1 receives two approvals and a_2 receives one approval. It is easy to see, that $\{a_2\}$ maximizes the cost-based satisfaction (with a value of ℓ), while $\{a_1\}$ is selected by the proportional greedy rule with a satisfaction of two. $\square$

To show correctness for Proposition 3.2 in [25] (i.e., all three hybrid rules modeling a $(1 - 1/\sqrt{e})$-approximation), we illustrate how each of the max rules coincides with a restricted version of the *budgeted maximum coverage* problem by Khuller, Moss, and Naor [106], who present according approximation results. Formally, the search problem BUDGETMAXCOVER is defined as follows.

Definition. *The* budgeted maximum coverage *problem, denoted* BUDGETMAXCOVER*, consists of a tuple $M = (\mathcal{S}, X, c, w, L)$, where $X = \{x_1, \ldots, x_n\}$ is a set of elements associated with a weight function $w : X \to \mathbb{N}$, $\mathcal{S} = \{S_1, \ldots, S_m\}$ is a collection of non-empty subsets of X associated with a cost function $c : \mathcal{S} \to \mathbb{N}$, and $L \in \mathbb{N}$ is a budget limit. The task is to output a set $\mathcal{S}' \subseteq \mathcal{S}$ with $\sum_{S' \in \mathcal{S}'} c(S') \leq L$, maximizing $\sum_{x' \in X'} w(x')$, where $X' = \bigcup_{S' \in \mathcal{S}'} S'$ is the set of elements covered by $\mathcal{S}$. Slightly abusing notation we also write $c(\mathcal{S}')$ and $w(\mathcal{S}')$, but explicitly point out, elements covered by multiple sets are only weighted once.*

By $\mathcal{M}(c, w, \lambda)$ we denote all BUDGETMAXCOVER *instances $(\mathcal{S}, X, c, w, L)$ where any element $x \in X$ is contained in at most λ sets. Excluding trivial instances, we define $1 \leq \lambda \leq n$. Slightly abusing notation we will denote unit cost and unit weight problems as $\mathcal{M}(1, w, \lambda)$ and $\mathcal{M}(c, 1, \lambda)$, implying that the given weight (respectively cost) function always maps to a constant value of one. Additionally, we denote by $\mathcal{M}(c, c, 1)$ the instances where the weight of any element is equivalent to the cost of its containing set.*

Subsequently, we will show that the max rules presented in Section 2 in [25] coincide with restricted domains of the BUDGETMAXCOVER problem. Here, equivalence means that the set of winning bundles according to the max rule equals the maximizing subsets in the corresponding budgeted maximum coverage instance. Note that the unit cost and unit weight problem $\mathcal{M}(1, 1, n)$ has been shown to be equivalent to the approval-based variant of the Chamberlin-Courant voting rule (see Chamberlin and Courant [50]) by Skowron and Faliszewski [159]. Since the budgeting method focusing on presence $\mathcal{R}^m_{\mathbb{1}_{|B_v|>0}}$ is a generalization of said voting rule, our approach will be quite similar and only differ in expanding both problems by a cost function.

Proposition. *The budgeting method $\mathcal{R}^m_{\mathbb{1}_{|B_v|>0}}$ coincides with the unit weight budgeted maximum coverage problem $\mathcal{M}(c, 1, n)$, while $\mathcal{R}^m_{|B_v|}$ coincides with $\mathcal{M}(c, 1, 1)$, and $\mathcal{R}^m_{c(B_v)}$ coincides with $\mathcal{M}(c, c, 1)$.*

Proof. For $\mathcal{R}^m_{\mathbb{1}_{|B_v|>0}}$ and any budgeting scenario $E = (A, V, c, \ell)$, we may construct a unit weight budgeted maximum coverage instance $M = (\mathcal{S}, V, c, w, \ell) \in \mathcal{M}(c, 1, n)$, where the collection $\mathcal{S} = \{S_1, \ldots, S_m\}$ represents the set of items $A = \{a_1, \ldots, a_m\}$ with $S_i = \{v \in V \mid a_i \in A_v\}$. Slightly abusing notation we imply identical costs $c(a_i) = c(S_i)$. Vice versa, we may also reverse above construction to construct any budgeting scenario E from any BUDGETMAXCOVER instance $M \in \mathcal{M}(c, 1, n)$. Slightly abusing notation we claim $\mathcal{R}^m_{\mathbb{1}_{|B_v|>0}}(E) = \text{BUDGETMAXCOVER}(M)$, where items in A and sets in $\mathcal{S}$ have a one-to-one relation. By construction, for every $B \subseteq A$ and $\mathcal{B} \subseteq \mathcal{S}$ with $a_i \in B \Leftrightarrow S_i \in \mathcal{B}$ it follows that $c(B) = c(\mathcal{B})$. Additionally, voter $v \in V$ is covered by $\mathcal{B}$ if and only if $B_v \neq \emptyset$. Equivalence follows, since each satisfied voter in E contributes to the total satisfaction by exactly a value of one, just as each covered element in M contributes to the total weight with a weight of exactly one. Since any bundle B and its counterpart $\mathcal{B}$ yield the exact same score in respective frameworks, B is a winning bundle in E if and only if $\mathcal{B}$ is a winning bundle in M.

For $\mathcal{R}^m_{|B_v|}$ and $\mathcal{R}^m_{c(B_v)}$ along with any budgeting scenario $E = (A, V, c, \ell)$, note that we can replace each voter $v \in V$ with $|A_v|$ voters, each approving exactly one element of A_v, without changing the overall satisfaction with any given bundle $B \subseteq A$. By restricting the construction for $\mathcal{R}^m_{\mathbb{1}_{|B_v|>0}}$ by assuming $\lambda = 1$, it follows easily, that $\mathcal{R}^m_{|B_v|}$ is equivalent to $\mathcal{M}(c, 1, 1)$. By additionally setting the weight function to the cost function, i.e., setting $w(v)$ to $c(\{v\})$ for all $v \in V$, we derive equivalence of $\mathcal{R}^m_{c(B_v)}$ to $\mathcal{M}(c, c, 1)$. □

We identified that all presented max rules are special cases of BUDGETMAXCOVER Hence, approximation algorithms for BUDGETMAXCOVER may be used for the max rules as well. We refer to an optimized $(1 - 1/e)$-approximation by Khuller, Moss, and Naor [106] for further reading. Alongside, the authors provide a $(1 - 1/\sqrt{e})$-approximation for BUDGETMAXCOVER, which compares a proportional greedy solution to the first iteration of a greedy algorithm. By construction, this result is inherited by our hybrid greedy rules.

Axiomatic Analysis

Now, let us provide formal proofs for the axiomatic analysis of the hybrid greedy rules. For ease of reading, we split the results across multiple propositions.

Limit Monotonicity

Proposition. *All three hybrid greedy rules violate limit monotonicity.*

Proof. First, let us demonstrate that it is be sufficient to present a counterexample for $\mathcal{R}^h_{c(B_v)}$ (while the results for the remaining hybrid greedy rules follow immediately). We replace each voter $v \in V$ with $c(A_v)$ substitutes, such that for each item $a \in A_v$ exactly $c(a)$ substitutes vote only for a. The overall score derived for any bundle and $|B_v|$ (or $\mathbb{1}_{|B_v|>0}$) coincides with the initial (cost-based) satisfaction for the unmodified election. Hence, a violation of limit monotonicity transfers from $\mathcal{R}^h_{c(B_v)}$ to $\mathcal{R}^h_{|B_v|}$ and $\mathcal{R}^h_{\mathbb{1}_{|B_v|>0}}$.

Now, consider a budgeting scenario $E = (A, V, c, \ell)$, with $A = \{a_1, a_2, a_3\}$, $c(a_1) = c(a_2) = 5$ and $c(a_3) = 6$, $V = \{v_1, v_2, v_3\}$, $A_{v_1} = A$, $A_{v_2} = \{a_1, a_3\}$ and $A_{v_3} = \{a_1\}$, and $\ell = 10$. It is easy to see that $\mathcal{R}^h_{c(B_v)}(E) = \{\{a_1, a_2\}\}$, as a_1 must be selected both by a greedy and proportional greedy rule. If the budget limit is increased to $\ell' = 11$, the only winning bundle is $\mathcal{R}^h_{c(B_v)}(E') = \{\{a_1, a_3\}\}$. □

Discount Monotonicity

Proposition. *All three hybrid greedy rules violate discount monotonicity.*

Proof. For $\mathcal{R}^h_{|B_v|}$ and $\mathcal{R}^h_{\mathbb{1}_{|B_v|>0}}$, consider a budgeting scenario E, consisting of four items $A = \{a_1, a_2, a_3, a_4\}$ with $c(a_1) = 5$, $c(a_2) = c(a_3) = 1$, and $c(a_4) = 3$, and a budget limit $\ell = 7$. A total of 15 voters approve a single item each, such that a_1 is approved by five voters, a_2 and a_3 are approved by three voters each and a_4 is approved by the remaining four voters. By construction each bundle B admits the same satisfaction under both scoring functions, i.e., $\sum_{v \in V} |B_v| = \sum_{v \in V} \mathbb{1}_{|B_v|>0}$. Now, the only winning bundle $\{a_1, a_2, a_3\}$ with a total satisfaction of 11 is a greedy solution. Yet, when reducing the cost of a_1 to $c(a_1) = 4$, the only winning bundle $\{a_2, a_3, a_4\}$ is a proportional greedy solution with a total satisfaction of ten (while the only greedy solution is $\{a_1, a_4\}$ with a satisfaction of only nine). Since the discounted item a_1 is only budgeted in the original scenario, discount monotonicity is violated.

For $\mathcal{R}^h_{c(B_v)}$, the counterexample for $\mathcal{R}^g_{c(B_v)}$, provided by Talmon and Faliszewski [166] in an extended version [165], also holds for the hybrid rule with cost-based satisfaction. □

Splitting Monotonicity

Proposition. *The hybrid rules $\mathcal{R}^h_{c(B_v)}$ and $\mathcal{R}^h_{\mathbb{1}_{|B_v|>0}}$ violate splitting monotonicity.*

Proof. To see that $\mathcal{R}^h_{c(B_v)}$ does not satisfy splitting monotonicity, consider a budgeting scenario E, consisting of three items a_1, a_2, and a_3 with $c(a_1) = 5$ and $c(a_2) = c(a_3) = 8$, a budget limit of $\ell = 8$, and three voters, two of them approving every item and one voter approving a_1 only. While for the given budgeting method $\{a_2\}$ is among the set of winning bundles, when splitting a_2 into two items with equal cost of four, the only winning bundle is $\{a_3\}$.

For $\mathcal{R}^h_{\mathbb{1}_{|B_v|>0}}$ we provide a larger counter example, presented as the following table. There are six items $A = \{a_1, \ldots, a_6\}$ and two sets of ten voters (each). The left side of the table shows ten voters, three of them approving a_1 and a_2, three approving a_2 and a_3, and four of them approving a_1, a_2, and a_3. Analogously, on the right side there is one voter approving a_4, one approving a_6, and respectively four voters each approving either $\{a_4, a_5\}$ or $\{a_5, a_6\}$. The budget limit is $\ell = 8$ and costs are listed in the table below.

	a_1	a_2	a_3	a_4	a_5	a_6	ℓ
$c(a_i)$	2	6	2	1	5	1	8
	3	3		1			
		3	3	4	4		
	4	4	4		4	4	
						1	

Note that $\{a_2, a_4, a_6\}$ is the only greedy solution and $\{a_4, a_6, a_1, a_3\}$ is the only proportional greedy solution. Since both bundles have a total satisfaction of twenty, we arbitrarily pick the greedy bundle containing a_2 for $\mathcal{R}^h_{\mathbb{1}_{|B_v|>0}}$. Subsequently, we may split a_2 into two pieces at equal cost of three. In the modified instance, there are two greedy solutions, both containing a split of a_2 and item a_5, resulting in a total satisfaction of eighteen. Concurrently, $\{a_4, a_6, a_1, a_3\}$ is still the only proportional greedy solution with strictly higher satisfaction of twenty. Obviously, the latter does not contain a split of a_2. $\square$

Proposition. *The hybrid rule $\mathcal{R}^h_{|B_v|}$ satisfies splitting monotonicity.*

To see that $\mathcal{R}^h_{|B_v|}$ satisfies splitting monotonicity, we will demonstrate a slightly stronger result. Subsequently, we will show that the following claim implies the above proposition.

Claim C1. *For any budgeting campaign $E = (A, V, c, \ell)$, let E' be a modified campaign, where a distinct item $a^* \in A$ is split into A^*. Then, for $x \in \{g, p\}$ and all $B_x \in \mathcal{R}^x_{|B_v|}(E)$ with $a^* \in B_x$ it holds that $B_x \setminus \{a^*\} \cup A^* \in \mathcal{R}^x_{|B_v|}(E')$.*

Proof. For both greedy rules, consider a winning bundle $B_x \in \mathcal{R}^x_{|B_v|}(E)$ with $a^* \in B_x$. There must be at least one tie-breaking order, such that B_x is the (resolute) winning bundle. When splitting a^* into A^*, we adapt the tie-breaking order by replacing a^* with A^* (where the relative order of items in A^* may be arbitrary). Note that each item in A^* is exactly as appealing as a^* itself (scoring a satisfaction of one for each approving voter). Hence, if we execute the greedy rule $\mathcal{R}^g_{|B_v|}$ on the modified election E' (given the modified tie-breaking order), items are added in the same order as in the original election, resulting in bundle $B_g \setminus \{a^*\} \cup A^*$. For the proportional greedy rule $\mathcal{R}^p_{|B_v|}$, the argument is similar. By splitting, the cost for each item in A^* is reduced from the original cost of item a^*. Therefore, each item in A^* is even more appealing with respect to a proportional selection method. Overall, items in A^* may be selected earlier, while the available budget limit cannot be exhausted to the point that later added items (from B_p) are prevented from being implemented (since $c(B_p) = c(B_p \setminus \{a^*\} \cup A^*)$ holds). □

Lemma. *Claim C1 implies splitting monotonicity for $\mathcal{R}^h_{|B_v|}$.*

Proof. Finally, we prove that the correctness of Claim C1 implies splitting monotonicity. Let $E = (A, V, c, \ell)$ be a budgeting campaign with $B_h \in \mathcal{R}^h_{|B_v|}(E)$ and $a^* \in B_h$. Note that B_h must have been chosen among a greedy solution $B_g \in \mathcal{R}^g_{|B_v|}(E)$ and a proportional greedy solution $B_p \in \mathcal{R}^p_{|B_v|}(E)$. If both B_g and B_p contain a^*, the claim obviously implies splitting monotonicity. Therefore, assume $B, B' \in \{B_g, B_p\}$ with $a^* \in B$ and $a^* \notin B'$ (by construction it follows that $B = B_h$). Following Claim C1, if a^* is replaced by A^*, then $B \setminus \{a^*\} \cup A^*$ is a winning bundle for one of the underlying (greedy or proportional greedy) rules. In order to violate splitting monotonicity, all outcomes aggregated by the other rule must yield a strictly higher score, without containing at least one item of A^*. If this is the case for the modified election, then this must have already been the case in the original election. Yet, we deduced that the score associated with B' cannot be higher than the score for B, because B was picked by the hybrid greedy rule. Hence, by contradiction splitting monotonicity must be satisfied. □

Merging Monotonicity

Proposition. *All three hybrid greedy rules violate merging monotonicity.*

Proof. For $\mathcal{R}^h_{|B_v|}$, consider a budgeting scenario with six items $A = \{a_1, \dots, a_6\}$, a cost function with $c(a_i) = 1$ for all $i \neq 6$ and $c(a_6) = 3$, a budget limit $\ell = 3$, and two voters $V = \{v_1, v_2\}$, where v_1 approves all items and v_2 approves a_6 only. Then,$\{a_6\}$ is the only winning budget aggregated by the greedy rule and $\{a_1, a_2, a_3\}$ is a winning budget aggregated by the proportional greedy rule. Due to a higher satisfaction, $\{a_1, a_2, a_3\}$ is a winning budget for $\mathcal{R}^h_{|B_v|}$. Now, when we merge a_1 and a_2 into a single item, the only winning bundle $\{a_3, a_4, a_5\}$ does not contain the merged item.

For $\mathcal{R}^h_{\mathbb{1}_{|B_v|>0}}$, consider a budgeting scenario with five items $A = \{a_1, \ldots, a_5\}$, $c(a_i) = 1$ for all $i \neq 5$ and $c(a_5) = 3$, a budget limit $\ell = 3$, and four voters $V = \{v_1, \ldots, v_4\}$ with $A_{v_1} = A_{v_2} = \{a_2, a_3, a_4, a_5\}$, $A_{v_3} = \{a_1, a_5\}$ and $A_{v_4} = \{a_1\}$. Then, it is easy to verify that $\{a_1, a_2, a_3\}$ is a winning bundle, while merging a_2 and a_3 into a single item, the only winning bundle is $\{a_1, a_4\}$.

For $\mathcal{R}^h_{c(B_v)}$, consider a budgeting scenario, such that $A = \{a_1, \ldots, a_5\}$ with $c(a_1) = 2$, $c(a_2) = c(a_3) = 3$, and $c(a_4) = c(a_5) = 4$, $V = \{v_1, v_2\}$ with $A_{v_1} = A$ and $A_{v_2} = \{a_1\}$, and $\ell = 8$. Note that there are multiple winning bundles, two of them being a greedy solution $B_g = \{a_4, a_5\}$ and a proportional greedy solution $B_p = \{a_1, a_5\}$, both having a total satisfaction of eight. Being part of a winning bundle, we may merge a_4 and a_5. Yet, the only winning bundle in the modified budgeting campaign is a proportional greedy solution $\{a_1, a_2, a_3\}$ with a total satisfaction of ten. $\square$

A.2 Distortion in Attribute Approval Committee Elections

In this section, we complement our publication [21], by providing missing proofs (that were omitted due to space constraints), as well as supplementary results. For the ease of reading, we assign alphanumerical labels only to those observations and theorems, which are referenced explicitly.

Results on Distortion

Let us formally establish tight bounds on the distortion for each pair of our proposed derivation methods and extended scoring functions. As a useful tool, let us first present two simple observations.

Observation O1. *For $E \in \mathcal{E}$, a voter $v_i \in V$, and any committee $W \subseteq C$ it holds that $f^{si}(b_i, W) \geq f^{co}(b_i, W) \geq f^{cc}(b_i, W)$. In case $|W| = 1$ holds, those scores coincide.*

Observation O2. *For each $E \in \mathcal{E}$ with $|V| = 1$, some individual scoring function $f^y \in \{f^{si}, f^{cc}, f^{co}\}$, and a committee $W \subseteq C$ it holds that $f^y_\Sigma(V, W) = f^y_{\min}(V, W)$.*

Candidate Approval Ballots

Theorem. *For f^{si}_{Σ}, f^{co}_{Σ}, f^{cc}_{Σ}, $f^{si}_{\min}$, $f^{co}_{\min}$ and $f^{cc}_{\min}$ the distortion associated with threshold-approvals ($\mathfrak{a}_\tau$) is unbounded for $\tau \neq 1$. This even holds for singlewinner attribute approval elections (i.e., $k = 1$) with only two candidates ($|C| = 2$), two categories ($d = 2$), and one voter ($|V| = 1$).*

Proof. For $\tau = 0$, the result follows immediately from the fact that $\mathcal{W}(\sigma_{\mathfrak{a}_0}(E), k) = \mathcal{P}_k(C)$ holds for all $E = (D, C, V) \in \mathcal{E}$. In particular, for some specific E, where only exactly one k-committee yields a positive score, the distortion becomes unbounded.

For $\tau = 2$, consider a singlewinner attribute approval election $E = (D, C, V)$ with $D = D^1 \times D^2$ and $D^j = \{a^j_1, a^j_2\}$ for $j \in [2]$, $C = \{c_1, c_2\}$, and $V = \{v_1\}$. Further let $a(c_1) = (a^1_1, a^2_2)$, $a(c_2) = (a^1_1, a^2_1)$ and $b_1 = (\{a^1_2\}, \{a^2_2\})$. Therefore it holds that $f^{\text{si}}(b_1, \{c_1\}) = \frac{1}{2}$ and $f^{\text{si}}(b_1, \{c_2\}) = 0$. Clearly, $\{\{c_1\}\} = F^{\text{si}}_{\Sigma}(E, 1)$ is the only winning committee. Yet, it holds that $\mathfrak{a}_2(b_1) = \emptyset$. Therefore, we may construct an election E', where the attribute vectors associated with c_1 and c_2 are interchanged, resulting in $\{\{c_2\}\} = F^{\text{si}}_{\Sigma}(E', 1)$. Hence, it holds that $\{c_2\} \in \mathcal{W}(\sigma_{\mathfrak{a}_2}(E), 1)$, but $f^{\text{si}}_{\Sigma}(V, \{c_2\}) = 0$. Following Observations O1 and O2, the result transfers to all of our scoring functions. For any $\tau > 2$, the proof can be adapted by extending the number of attribute categories. $\square$

For $\tau = 1$, the distortion is bounded by the number of the attribute domains, as shown in the subsequent theorems.

Theorem T1. *For f^{si}_{Σ}, f^{co}_{Σ} and f^{cc}_{Σ}, the distortion associated with threshold-approvals ($\mathfrak{a}_\tau$) is in $\Theta(d)$ for $\tau = 1$.*

Proof. Let us first establish the upper bound $\mathcal{O}(d)$ for all three scoring functions. Consider any two attribute approval elections $E = (D, C, V) \in \mathcal{E}$ and $E' = (D', C, V') \in \mathcal{E}$ with $\mathfrak{a}_1(E) = \mathfrak{a}_1(E')$. Further, for any extended scoring function $f_\Sigma \in \{f^{\text{si}}_{\Sigma}, f^{\text{co}}_{\Sigma}, f^{\text{cc}}_{\Sigma}\}$, let $W \in F_\Sigma(E, k)$ and $W' \in F_\Sigma(E', k)$.

We start by exploring f^{si}_{Σ}. Let each voter $v_i \in V$ with ballot b_i be represented by a voter $v'_i \in V'$ with ballot b'_i. As E and E' induce equivalent candidate approval ballots, if derived from $\mathfrak{a}_1$, exactly one of the following conditions holds for every candidate $c \in C$ and every pair of voters v_i, v'_i. Either $c \notin \mathfrak{a}_1(b_i) \cup \mathfrak{a}_1(b'_i)$ or $c \in \mathfrak{a}_1(b_i) \cap \mathfrak{a}_1(b'_i)$, implying that either $f^{\text{si}}(b_i, \{c\}) = f^{\text{si}}(b'_i, \{c\}) = 0$ or both $f^{\text{si}}(b_i, \{c\}) > 0$ and $f^{\text{si}}(b'_i, \{c\}) > 0$. Having a committee W' that yields a maximum score for V', we can determine a lower bound for the overall score that voters V associate with W', i.e., $f^{\text{si}}_{\Sigma}(V, W')$ is at least:

$$\begin{aligned}
&\sum\nolimits_{v'_i \in V'} |\{c \in W' \mid f^{\text{si}}(b'_i, \{c\}) > 0\}|/d \\
=&\sum\nolimits_{v_i \in V} |\mathfrak{a}_1(b_i) \cap W'|/d
\end{aligned}$$

In contrast, an upper bound on the score for $W \in F_{\Sigma}^{\text{si}}(E,k)$ is given by $f_{\Sigma}^{\text{si}}(V,W) \leq \sum_{v_i \in V} |\mathfrak{a}_1(b_i) \cap W|$, bounding the distortion by d.

For $f_{\Sigma}^{\text{cc}}(V,W')$ and $f_{\Sigma}^{\text{cc}}(V,W)$, a voter's satisfaction with a committee that contains a candidate she approves is at least $1/d$ and at most one. On the other hand, a voter who is not represented at all in the outcome yields a satisfaction of zero. We derive a lower bound on the satisfaction of V with W' and an upper bound on the score for $W \in F_{\Sigma}^{\text{si}}(E,k)$ as follows, with a distortion of at most d:

$$f_{\Sigma}^{\text{cc}}(V,W') \geq |\{v_i \in V \mid \mathfrak{a}_1(b_i) \cap W' \neq \emptyset\}|/d$$
$$f_{\Sigma}^{\text{cc}}(V,W) \leq |\{v_i \in V \mid \mathfrak{a}_1(b_i) \cap W| \neq \emptyset\}|$$

We can determine lower and upper bounds for $f_{\Sigma}^{\text{co}}(V,W')$ and $f_{\Sigma}^{\text{co}}(V,W)$ analogously. As for f_{Σ}^{cc} the satisfaction for a voter with a given committee is at least $1/d$ and at most one if at least one preferred candidate is present, and zero otherwise. Using the same reasoning as before, the distortion is at most d.

For the lower bound $\Omega(d)$, consider $E^* = (D,C,V) \in \mathcal{E}$ with one voter $V = \{v_1\}$, two candidates $C = \{c_1, c_2\}$ and $k = 1$. Following Observation O2, it is sufficient to argue for f_{Σ}^{si}. From the perspective of v_1, let c_1 satisfy one category and c_2 every category, i.e., $f^{\text{si}}(b_1, \{c_1\}) = 1/d$ and $f^{\text{si}}(b_1, \{c_2\}) = 1$. Then $\{c_2\}$ is a winning committee with $f_{\Sigma}^{\text{si}}(V, \{c_2\}) = 1$. It holds that $\mathfrak{a}_1(b_1) = C$ and we can construct a modified election, such that c_1 satisfies all the categories and c_2 satisfies only one category. In this modified election, the only winning committee is $\{c_1\}$, but v_1 approves all candidates in both candidate approval elections (derived using $\mathfrak{a}_1$). Overall it holds that $\{c_1\} \in \mathcal{W}(\sigma_{\mathfrak{a}_1}(E^*), 1)$ with score $f_{\Sigma}^{\text{si}}(V, \{c_1\}) = 1/d = f_{\Sigma}^{\text{si}}(V, \{c_2\})/d$. Thus, the distortion is at least d. $\square$

Theorem. *For $f_{\min}^{si}$, $f_{\min}^{co}$ and $f_{\min}^{cc}$, the distortion associated with threshold-approvals ($\mathfrak{a}_\tau$) with $\tau = 1$ is in $\Theta(d)$.*

Proof. The proof builds on the previous proof of Theorem T1 and is therefore only sketched. Again, we begin with the upper bound. For $f_x^y \in \{f_{\min}^{\text{co}}, f_{\min}^{\text{cc}}\}$, note that the satisfaction of the voters V with a committee W' is either zero (if at least one voter is dissatisfied) or greater than zero. By the arguments presented in the previous proof, if W' is a winning committee in an election E', then W' has a positive score if and only if W' also yields a positive score in E with $\mathfrak{a}_1(E) = \mathfrak{a}_1(E')$. Now, for a fixed voter, consider the bounds on the scores for committees, which yield a score greater than zero. For $f_{\min}^{\text{co}}$, and $f_{\min}^{\text{cc}}$ the minimum score is $1/d$ and the maximum score is 1, yielding a distortion of at most d. For $f_{\min}^{\text{si}}$ and any voter, the number of approved candidates in W' divided by d is a lower bound on the score of any winning committee in E. Again, this score can be at most d times higher, if the candidates in W' only satisfy one attribute for the least satisfied voter.

For the lower bound, the election E^* from the previous proof of Theorem T1 can be used for $f^{\text{si}}_{\min}$, $f^{\text{co}}_{\min}$ and $f^{\text{cc}}_{\min}$ following Observation O1. $\square$

Cardinal Preferences

For each $b \in \mathcal{D}$ and $c \in C$, it holds that $\mathfrak{c}(b, c) = f^{\text{si}}(b, \{c\}) = f^{\text{cc}}(b, \{c\})$ by definition. Thus, we derive the following observation.

Observation. *The distortion for cardinal preferences, derived using method $\mathfrak{c}$, along with f^{si}_{Σ}, $f^{si}_{\min}$, f^{cc}_{Σ} and $f^{cc}_{\min}$ is one.*

Yet, the score for f^{co} cannot be broken down to single candidates. Hence, we investigate the distortion for cardinal preferences (derived by using $\mathfrak{c}$), paired with f^{co}_{Σ} or $f^{\text{co}}_{\min}$.

Theorem T2. *The distortion for f^{co}_{Σ} and cardinal preferences ($\mathfrak{c}$) is in $\Theta(\min(k, d))$.*

Proof. We start with the upper bound $\mathcal{O}(\min(k, d))$. Consider any attribute approval election $E = (D, C, V) \in \mathcal{E}$. Note that for any voter $v_i \in V$, the satisfaction with a given committee $W' \in \mathcal{P}_k(C)$ is bound downwards by

$$f^{\text{co}}(b_i, W') \geq \max_{c \in W'} \mathfrak{c}(b_i, c). \tag{EQ1}$$

The intuition is, that for v_i there can be a candidate $c \in W'$, who satisfies the most categories and all the other candidates $c' \in W' \setminus \{c\}$ only satisfy attributes in the same categories as c. Let $E' = (D', C, V') \in \mathcal{E}$ be an attribute approval election with $\mathfrak{c}(E) = \mathfrak{c}(E')$. For any winning committee $W \in F^{\text{co}}_{\Sigma}(E, k)$ in the initial election E, we can deduce a lower bound ($t \in \mathbb{Q}$) for the score of any winning committee in E' using Equation (EQ1):

$$f^{\text{co}}_{\Sigma}(V', W') \geq f^{\text{co}}_{\Sigma}(V', W) \geq \sum_{v_i \in V} \max_{c \in W} \mathfrak{c}(b_i, c) = t$$

Therefore, any winning committee $W' \in F^{\text{co}}_{\Sigma}(E', k)$ with respect to E' has a score of at least t. Using the same argument from Equation (EQ1) again in the other direction, it follows that $f^{\text{co}}_{\Sigma}(V, W') \geq t$. Hence, the distortion for f^{co}_{Σ} and election E is at most

$$\text{dist}(\mathfrak{c}, f^{\text{co}}_{\Sigma}, E) \leq \max_{W \in F^{\text{co}}_{\Sigma}(E,k)} \frac{\sum_{v_i \in V} \min(\sum_{c \in W} \mathfrak{c}(b_i, c), 1)}{\sum_{v_i \in V} \max_{c \in W} \mathfrak{c}(b_i, c)}. \tag{EQ2}$$

The denominator follows directly from our previous steps. The intuition for the numerator is, that for any voter $v_i \in V$ the satisfaction with a given committee is capped in two different ways. It is *(i)* capped by one, i.e., if a maximizing committee W satisfies every category for v_i, and it is *(ii)* capped by $\sum_{c \in W} \mathfrak{c}(b_i, c)$, i.e., the theoretical maximum of satisfied categories, if no two candidates $c, c' \in W$ satisfy a joint category for v_i (divided by d for normalization).

Finally, we can determine an upper bound on the distortion based on given parameters. If $d \leq k$, the distortion is maximal for an election E, where each voter approves of exactly one attribute for every candidate of two different committees W, W', while the satisfied attributes overlap in W', but complement each other in W. The resulting distortion is $\mathrm{dist}(\mathfrak{c}, f^{\mathrm{co}}_{\Sigma}, E) \leq d$. For $d > k$, the distortion is maximal in an election, where each voter approves of at most $\lambda \leq \lfloor d/k \rfloor$ attributes for every candidate of two committees W, W' in a similar way (i.e., covering W and overlapping in W'). In this case the distortion is given by $\mathrm{dist}(\mathfrak{c}, f^{\mathrm{co}}_{\Sigma}, E) \leq \frac{\lambda k/d}{\lambda/d} \leq k$.

For the lower bound $\Omega(\min(k, d))$, we construct an attribute approval election $E^* = (D, C, V) \in \mathcal{E}$ as follows. Let $d = k$, $C = \{c_1, \dots, c_{2k}\}$ and each candidate c_i has a unique attribute c_i^j for all $j \in [d]$. Let $V = \{v\}$, such that there is only one voter with ballot $b = (B^1, \dots, B^k) \in \mathcal{D}$. For a distinct k-committee $W = \{c_1, \dots, c_k\}$, we construct b in a way, such that for each $j \in [1, k-1]$ it holds that $B^j = \{c_j^j\}$ and $B^k = \bigcup_{i \in [k,2k]} \{c_i^k\}$. That is, with respect to the ballot b, every candidate satisfies exactly one attribute, but $f^{\mathrm{co}}(V, W) = 1$ and $f^{\mathrm{si}}(V, C \setminus W) = 1/d$. Obviously, we can construct a similar election E' with $\mathfrak{c}(E^*) = \mathfrak{c}(E')$, where the opposite holds, such that $C \setminus W$ is a winning committee. Overall the distortion $\mathrm{dist}(\mathfrak{c}, f^{\mathrm{co}}_{\Sigma}, E^*)$ is at least $d = k = \min(k, d)$. □

Theorem. *The distortion for $f^{\mathrm{co}}_{\min}$ and cardinal preferences ($\mathfrak{c}$) is in $\Theta(\min(k, d))$.*

Proof. The proof follows directly by modifying the previous proof of Theorem T2. For the upper bound the result also holds by replacing the sum operator in Equation (EQ2), resulting in $\mathrm{dist}(\mathfrak{c}, f^{\mathrm{co}}_{\min}, E)$ being at most

$$\max_{W \in F^{\mathrm{co}}_{\min}(E,k)} \frac{\min_{v_i \in V} \min(\sum_{c \in W} \mathfrak{c}(b_i, c), 1)}{\min_{v_i \in V} \max_{c \in W} \mathfrak{c}(b_i, c)}.$$

The case study for $d \leq k$ and $d > k$ is also very similar to the previous proof. First note, that in an election E with maximal distortion, there cannot exist a voter which is completely dissatisfied with a winning committee (since we are considering the $\min$ operator, resulting in a satisfaction of zero in both elections). By the same arguments presented in the previous proof, there is also no unsatisfied voter in any winning committee of an election $E' \in \sigma_{\mathfrak{c}}(E)$. More precisely, $f^{\mathrm{co}}_{\min}(V, W') \geq 1/d$ for all $W' \in \mathcal{W}(\sigma_{\mathfrak{c}}(E))$.

Now, for the case study consider the least satisfied voter $v \in V$ for an election $E = (D, C, V)$. For $d \leq k$ it is easy to see, that the distortion can be at most d, since $f^{\mathrm{co}}_{\min}(V, W) \leq 1$ for all $W \in \mathcal{P}_k(C)$. For $d > k$, let $W' \in \mathcal{W}(\sigma_{\mathfrak{c}}(E))$ be a committee, such that $f^{\mathrm{co}}_{\min}(V, W')$ is minimal. We know that

$$f^{\mathrm{co}}_{\min}(V, W') \geq \min_{v_i \in V} \max_{c \in W'} \mathfrak{c}(b_i, c) = t,$$

which is the number of approved attributes, the worst off voter assigns to her best candidate. In a different committee W, the satisfaction for the worst off voter, can be at most

$t \cdot k$, if the approved attributes overlap in W', but complement each other in W. Latter results in a distortion of at most k. The overall distortion is in $\mathcal{O}(\min(k, d))$ for $f^{\mathrm{co}}_{\min}$.

Following Observation O2, the lower bound result for f^{co}_{Σ} in the proof of Theorem T2 also holds for f^{co}_{Σ}, i.e., E^* also yields a distortion of at least $\min(k, d)$ for f^{co}_{Σ}. □

Ordinal Preferences

Theorem T3. *The distortion for f^{si}_{Σ} and ordinal preferences ($\mathfrak{o}$) is in $\Theta(d)$.*

Proof. For the upper bound, consider $E = (D, C, V)$ and $E' = (D', C, V')$ with $\mathfrak{o}(E) = \mathfrak{o}(E')$ and let $W \in F^{\mathrm{si}}_{\Sigma}(E, k)$ and $W' \in F^{\mathrm{si}}_{\Sigma}(E', k)$. We construct a suitable lower bound on the satisfaction for W' (perceived by V) of $f^{\mathrm{si}}_{\Sigma}(V, W') \geq \ell_1 + \ell_2$, in two steps. First, for every voter $v_i \in V$, any candidate $c \in C$ yields a score of at least $1/d \cdot \sum_{j\in[d]} |B^j_i \cap \{c^j\}|$, which is bounded downwards by the number of attributes satisfied by the least favorite candidate (which admits the lowest score). That is,

$$\ell_1 = \frac{k}{d} \cdot \sum_{v_i \in V} \min_{c\in C} \sum_{j\in[d]} |B^j_i \cap \{c^j\}|.$$

Secondly, for $v_i \in V$, candidates that are not covered by ℓ_1 (candidates not appearing last in the linear ranking) yield an additional score of at least $1/d$, i.e., at least one additional attribute must be satisfied. Hence, we derive

$$\ell_2 = \frac{1}{d} \sum_{v_i \in V} |\{c' \in W' \mid \exists c \in C \text{ with } c' \succ_{b_i} c\}|.$$

The intuition is, that every candidate, which is not at the last position of a linear ranking, must admit at least $1/d$ points. This lower bound $\ell = \ell_1 + \ell_2$ holds for any election $E' \in \mathcal{E}$ with $\mathfrak{o}(E) = \mathfrak{o}(E')$.

In contrast, a generous upper bound is $f^{\mathrm{si}}_{\Sigma}(V, W) \leq \ell_1 + d \cdot \ell_2$, i.e., candidates positioned last yield a total score of at least ℓ_1, while candidates that are not at the last position of a ranking, can at most admit d/d additional points, each. Overall, for any E, E' with $\mathfrak{o}(E) = \mathfrak{o}(E')$, and $W \in F^{\mathrm{si}}_{\Sigma}(E, k)$ and $W' \in F^{\mathrm{si}}_{\Sigma}(E', k)$, it holds that

$$\mathrm{dist}(\mathfrak{o}, f^{\mathrm{si}}_{\Sigma}, E) \leq \frac{f^{\mathrm{si}}_{\Sigma}(V, W)}{f^{\mathrm{si}}_{\Sigma}(V, W')} \leq \frac{\ell_1 + \ell_2 \cdot d}{\ell_1 + \ell_2} \leq d.$$

For the lower bound, consider an election $E^* \in \mathcal{E}$ with two voters $V = \{v_1, v_2\}$, two candidates $C = \{c_1, c_2\}$, and $k = 1$. Further, let c_1 and c_2 not share any attribute. We construct the voters' ballots b_1 and b_2 in a way, that $\mathfrak{c}(b_1, c_1) = 1/d$, $\mathfrak{c}(b_1, c_2) = 0$, $\mathfrak{c}(b_2, c_1) = 0$, and $\mathfrak{c}(b_2, c_2) = 1$. Clearly it holds that $\{c_2\}$ is a winning committee with

$f^{\mathrm{si}}_{\Sigma}(V, \{c_2\}) = 1$. On the other hand, consider an election $E' = (D, C, V')$. This time, the voters' ballots b'_1 and b'_2 are swapped, i.e., constructed by setting $\mathfrak{c}(b'_i, c_j) = \mathfrak{c}(b_{3-i}, c_{3-j})$ for $i, j \in [2]$. Note that $\mathfrak{o}(E^*) = \mathfrak{o}(E')$ holds, but $\{c_1\}$ is a winning committee in E' with $f^{\mathrm{si}}_{\Sigma}(V, \{c_1\}) = 1/d$. Hence, $\mathrm{dist}(\mathfrak{o}, f^{\mathrm{si}}_{\Sigma}) \geq \mathrm{dist}(\mathfrak{o}, f^{\mathrm{si}}_{\Sigma}, E^*) = d$. □

Theorem. *The distortion for f^{cc}_{Σ} and ordinal preferences ($\mathfrak{o}$) is in $\Theta(d)$.*

Proof. The proof is similar to the previous proof of Theorem T3. The election E^* used for the lower bound also yields a distortion of at least d, following Observation O1.

For the upper bound, we consider the elections E and E' from the previous proof. Again, we obtain a lower bound $\ell \leq f^{\mathrm{cc}}_{\Sigma}(V, W')$, consisting of $\ell = \ell_1 + \ell_2$, which for f^{cc}_{Σ} and some $W' \in F^{\mathrm{cc}}_{\Sigma}(E', k)$ are given by

$$\ell_1 = \frac{1}{d} \cdot \sum_{v_i \in V} \min_{c \in C} \sum_{j \in [d]} |B^j_i \cap \{c^j\}| \text{ and}$$
$$\ell_2 = \frac{1}{d} |\{v_i \in V \mid \exists c' \in W' \text{ and } \exists c \in C \text{ with } c' \succ_{b_i} c\}|.$$

Overall, ℓ_1 is the sum of the scores of all voters, any candidate will yield at the last position of respective ordinal rankings. On the other hand, ℓ_2 is the number of voters, that prefer at least one candidate in W' to at least one other candidate in their ordinal ranking (divided by d). Latter models, that a candidate not ranked last must yield an additional score of at least $1/d$. Again, a generous upper bound is given by $f^{\mathrm{cc}}_{\Sigma}(V, W) \leq \ell_1 + d \cdot \ell_2$. We derive a distortion of at most d. □

For the remaining scoring functions, the distortion is unbounded.

Theorem. *For distortion f^{co}_{Σ} and ordinal preferences ($\mathfrak{o}$) is unbounded.*

Proof. Consider an attribute approval election $E = (D, C, V) \in \mathcal{E}$ with $k = 3$, $d = 3$, and $C = \{c_1, \ldots, c_5\}$ such that every candidate c_i has a unique attribute c^j_i in each category $j \in [3]$. There is only one voter $V = \{v_1\}$, casting ballot $b_1 \in \mathcal{D}$ with $b_1 = (\{c^1_1\}, \{c^2_2\}, \{c^3_1, c^3_2\})$, inducing the following ordinal preference:

$$\mathfrak{o}(b_1) : c_1 \sim_{b_1} c_2 \succ_{b_1} c_3 \sim_{b_1} c_4 \sim_{b_1} c_5$$

Obviously, it holds that $W = \{c_1, c_2, c_3\}$ is a winning committee with $f^{\mathrm{co}}_{\Sigma}(V, W) = 1$. We construct $E' \in \mathcal{E}$ with $\mathfrak{o}(E) = \mathfrak{o}(E')$, by only replacing the set of voters with $V' = \{v'_1\}$, casting the ballot $b'_1 = (\{c^1_1, c^1_2, c^1_3\}, \{c^2_1, c^2_2, c^2_4\}, \{c^3_5\})$. Note that $\mathfrak{o}(b'_1) = \mathfrak{o}(b_1)$ and $W' = \{c_3, c_4, c_5\}$ is a winning committee in E' with $f^{\mathrm{co}}_{\Sigma}(V', W') = 1$. Yet, in the initial election E it holds that $f^{\mathrm{co}}_{\Sigma}(V, W') = 0$. Hence, the distortion is unbounded. □

Theorem. *For $f^{si}_{\min}$, $f^{co}_{\min}$ and $f^{cc}_{\min}$ the distortion associated with ordinal preferences, derived by $\mathfrak{o}$, is unbounded, even if restricted to singlewinner elections (i.e., $k = 1$) with $d = 2$, $|C| = 2$, and $|V| = 2$.*

Proof. We modify the lower bound proof of Theorem T3. In particular, in election E^* we only consider the first $d = 2$ categories and modify the voters' ballots, such that $\mathfrak{c}(b_1, c_1) = 1$, $\mathfrak{c}(b_1, c_2) = 1/2$, $\mathfrak{c}(b_2, c_1) = 0$ and $\mathfrak{c}(b_2, c_2) = 1/2$. The resulting ordinal preferences are given by $\mathfrak{o}(b_1) : c_1 \succ_{b_1} c_2$ and $\mathfrak{o}(b_2) : c_2 \succ_{b_2} c_1$. E' is constructed by the same method as in the proof of Theorem T3, by approving different attributes without changing the ordinal preferences of the voters. Finally, $\{c_2\}$ is a winning committee in E^* and $\{c_1\}$ is a winning committee in E'. Yet, $f^{\text{si}}(b_2, \{c_1\}) = 0$ and $f^{\text{si}}(b_i, \{c_2\}) > 0$ for $i \in [2]$. Extending the individual scoring to $f^{\text{si}}_{\min}$, it holds that $f^{\text{si}}_{\min}(V, \{c_2\}) > 0$ and $f^{\text{si}}_{\min}(V, \{c_1\}) = 0$, resulting in unbounded distortion. Following Observation O1, the proof also holds for $f^{\text{co}}_{\min}$ and $f^{\text{cc}}_{\min}$. □

Extension to Weighted Attribute Approval Elections

In this section, let us illustrate how our results on distortion formally extend to weighted elections. We consider two cases. Either each category $j \in [d]$ is associated with a *global* weight $w^j \in \mathbb{Q}_{\geq 0}$, such that $0 \leq w^j \leq 1$ and $\sum_{j \in [d]} w^j = 1$, or for a voter $v_i \in V$, each $j \in [d]$ is associated with an *individual* weight $w_i^j \in \mathbb{Q}_{\geq 0}$, satisfying analogous conditions. Our results can be transferred to weighted elections, by encoding the weights into the elections. This can be done by duplicating categories and adapting the candidates and voters accordingly. Formally, we simply encode weights into an election $E = (D, C, V) \in \mathcal{E}$ as follows.

For global weights, let $\lambda \in \mathbb{N}_+$ be the smallest value, such that for all $j \in [d]$, it holds that $\lambda \cdot w^j \in \mathbb{N}_0$ is a non-negative integer. We construct $E' \in \mathcal{E}$, by duplicating respective attribute categories. That is, for $w^j = 0$, we remove the j-th category D^j from D (and also from the candidates' attribute vectors and the voters' ballots). For $w^j > 0$, we replace D^j with $\lambda \cdot w^j$ identical clones (again, also for C and V). Extending derivation methods accordingly, the distortion scales by replacing d with $\lambda \cdot d$ for the lower bound results (assuming there is at least one weight w^j with $\lambda \cdot w^j = 1$). Considering some arbitrarily small weight $w^j < \epsilon$, the distortion for almost all rules and all derivation methods can become arbitrarily bad.

For individual weights, recall that we assume normalization, i.e., for a voter $v_i \in V$, each $j \in [d]$ is associated with an individual weight w_i^j, such that $0 \leq w_i^j \leq 1$ and $\sum_{j \in [d]} w_i^j = 1$. We can extend an election $E = (D, C, V) \in \mathcal{E}$ in a similar way. Let $\lambda \in \mathbb{N}_+$ be the smallest value such that $\lambda \cdot w_i^j \in \mathbb{N}_0$ holds for all $v_i \in V$ and all $j \in [d]$. This time we construct $E' \in \mathcal{E}$, by replacing every category D^j by λ identical clones and extend respective candidates. For the set of voters, let v_i cast a modified ballot b_i' as follows. If B_i^j are the attributes voter v_i approves of in the j-th category in the original ballot b_i, then the approving set B_i^j is replaced in b_i' with a combination of $\lambda \cdot w_i^j$ clones of B_i^j and $\lambda \cdot (1 - w_i^j)$ empty sets (i.e., not approving any attribute). Informally, this relates to replacing each category by λ clones (resulting in $\lambda \cdot d$ categories overall), while each voter is allowed to pose λ non-empty entries in her ballot. For normalization, the scoring functions can be slightly altered, by multiplying respective scores by the number of categories in the unweighted election d.

Expressiveness

Lastly, we investigate how expressive attribute approval ballots are. In particular, attribute approval ballots (as well as candidate approvals, cardinal preferences, and ordinal preferences) induce a weak linear order over the set of k-committees, which also depends on an underlying individual scoring function. For example, if a voter's satisfaction with a committee is measured by the presence of at least one approved candidate, any candidate approval ballot induces a dichotomous preference over all k-committees. We are interested in how well attribute approval ballots are able to model arbitrary rankings over fixed-size committees, given that there is a sufficiently large number of unique attributes for the candidates. The results in this section are formally about votes and individual scoring functions. In order to simplify notation, we will present them in the introduced concept of elections with a focus on a particular voter v_i. Further, for a given set C, let $\omega(\mathcal{P}_k(C))$ be the collection of all weak rankings over the set of k-committees $\mathcal{P}_k(C)$.

Definition. *Let $(D, C, V) \in \mathcal{E}$ be an election, f an individual scoring function, and b_i an attribute approval ballot of some voter $v_i \in V$. Then $\succsim_i^f \in \omega(\mathcal{P}_k(C))$ denotes the weak linear order induced by f and b_i, such that for every two committees $W, W' \in \mathcal{P}_k(C)$ it holds that $W \succ_i^f W' \Leftrightarrow f(b_i, W) > f(b_i, W')$ and $W \sim_i^f W' \Leftrightarrow f(b_i, W) = f(b_i, W')$. If clear from context, f is omitted.*

Simple Scoring and Other Additive Scoring Functions

Theorem T4. *For f^{si}, any candidate set C, and $k \in \{1, m-1, m\}$ there is an attribute approval election $(D, C, V) \in \mathcal{E}$, such that for every weak ranking $\succsim \in \omega(\mathcal{P}_k(C))$, there exists a ballot $b_i \in \mathcal{D}$, inducing $\succsim$ (i.e., $\succsim_i = \succsim$). For $k \in \{2, \dots, m-2\}$, there is no such election and ballot for any additive individual scoring function f.*

Proof. We study four cases. For $k = m$, the only k-committee is C. Trivially, there is only one order over sets with one element.

For $k = 1$, consider an election (D, C, V) with $d = m$ categories. Further, let each category $j \in [m]$ have (at least) m attribute specifications $a_1^j, \dots, a_m^j$, such that each candidate has an unique attribute in each category, i.e., $c_i^j = a_i^j$ for all $i, j \in [m]$. To induce a weak ranking $\succsim$ over 1-committees, we construct a ballot $b_i = (B_i^1, \ldots, B_i^m)$ as follows. For each candidate $c \in C$, let $t(c) = |\{c' \in C \mid \{c\} \succsim \{c'\}\}|$ be the number of 1-committees (i.e., candidates), $\{c\}$ is weakly preferred to in $\succsim$. Then for every $c \in C$, ballot b_i shares exactly $t(c)$ attributes with c, e.g., for each $j \in [t]$ it holds that $c^j \in B_i^j$. Note that $t(c)$ models the *Borda* score (see [177, 35, 90]) for each candidate $c \in C$. More precisely, with respect to the ballot b_i, c satisfies exactly the number of attributes equivalent to its Borda score in $\succsim$. Hence, $\succsim_i = \succsim$ holds by construction.

For $k = m-1$, we can build on the previous construction. Imagine any $(m-1)$-committee $C' \in \mathcal{P}_{m-1}(C)$ as $C' = C \setminus \{c\}$ for some distinct candidate c. Hence, there is a one-to-one relationship between a ranking $\succsim' \in \omega(\mathcal{P}_{m-1}(C))$ and a ranking $\succsim \in \omega(\mathcal{P}_1(C))$. We simulate the removal of a candidate by assigning each candidate a value of m minus her Borda score. Then, we have $b_i' = (B_i^{1'}, \ldots, B_i^{m'})$ such that for all $j \in [m]$ it holds that $B_i^{j'} = D^j \setminus B_i^j$. Note that for $c, c' \in C$ with $C \setminus \{c\} \succsim' C \setminus \{c'\}$ it holds that $f^{\text{si}}(b_i', C \setminus \{c\}) = f^{\text{si}}(b_i', C) - (m - t(c)) \geq f^{\text{si}}(b_i', C) - (m - t(c')) = f^{\text{si}}(b_i', C \setminus \{c'\})$, resulting in $\succsim_i = \succsim'$.

For $k \notin \{1, m-1, m\}$ and any additive individual scoring function f, there is always a weak linear ranking $\succsim \in \omega(\mathcal{P}_k(C))$ over a set of candidates C, such that there is no election $(D, C, V) \in \mathcal{E}$ and ballot $b_i \in \mathcal{D}$ with $\succsim_i = \succsim$. Consider the following counterexample with $m = 4$ and $k = 2$. For $C = \{a, b, x, y\}$ assume there is a ballot b_i, inducing $\succsim_i = \succsim$ with

$$\succsim\colon \{a, b\} \succ \{a, x\} \sim \{a, y\} \sim \{b, x\} \sim \{b, y\} \sim \{x, y\}.$$

By additivity for f, it follows that $f(b_i, \{b\}) > f(b_i, \{x\})$ from $\{a, b\} \succ_i \{a, x\}$, but also $f(b_i, \{b\}) = f(b_i, \{x\})$ from $\{b, y\} \sim_i \{x, y\}$, which is a contradiction. The example can easily be extended by adding additional m' candidates and increasing k by at most m'. Still, there is no ballot b_i such that $\succsim_i$ models a preference, such that exactly one committee is preferred to all other committees. □

Chamberlin-Courant Scoring

Theorem. *For f^{cc}, any candidate set C, and $k \in \{1, m\}$ there is an attribute approval election $(D, C, V) \in \mathcal{E}$, such that for every weak ranking $\succsim \in \omega(\mathcal{P}_k(C))$, there exists a ballot $b_i \in \mathcal{D}$, inducing $\succsim$ (i.e., $\succsim_i = \succsim$). For $k \in \{2, \ldots, m-2\}$ and f^{cc}, there is no such election. and ballot.*

Proof. The cases $k = m$ and $k = 1$ follow for the same reasons as described in the proof of Theorem T4. For $k \notin \{1, m\}$ we construct a counterexample $(D, C, V) \in \mathcal{E}$ with $C = \{a, b, c\}$ and $k = 2$. Let $\succsim_i = \succsim$ be a strict linear ranking over 2-committees, e.g.,

$$\succsim\colon \{a, b\} \succ \{b, c\} \succ \{a, c\}.$$

For some ballot b_i to induce $\succsim_i$, it holds that $\{a,b\} \succ_i \{b,c\}$ implies $f^{\text{cc}}(b_i,\{a\}) > f^{\text{cc}}(b_i,\{b\})$, while $\{b,c\} \succ_i \{a,c\}$ implies $f^{\text{cc}}(b_i,\{b\}) > f^{\text{cc}}(b_i.\{a\})$, resulting in a contradiction. Clearly, we can extend this counterexample to also hold for $m > 3$ and any $k \notin \{1,m\}$. □

Committee Scoring

Theorem. *For f^{co}, any candidate set C, and any weak ranking $\succsim \in \omega(\mathcal{P}_k(C))$, there is an attribute approval election (D,C,V) and a ballot $b_i \in \mathcal{D}$ with $\succsim = \succsim_i$.*

Proof. For any fixed set of candidates $C = \{c_1,\dots,c_m\}$ and any committee size k, the number of k-committees is a constant $t = |\mathcal{P}_k(C)| = \binom{m}{k}$. We construct (D,C,V) and assume there is a finite, but sufficiently large, number of categories d. Further, in each category D^j for $j \in [d]$, there are m attribute specifications, such that each candidate has a unique property in each category, i.e., $\bigcup_{c \in C}\{c^j\} = D^j$. As an intermediate step, we construct a ballot b_i, such that for a specific k-committee $W \in \mathcal{P}_k(C)$, it holds that $f^{\text{co}}(b_i,W) = \frac{t-1}{d}$ and for all $W' \in \mathcal{P}_k(C)$ with $W' \neq W$ it holds that $f^{\text{co}}(b_i,W') = \frac{t-2}{d}$. That is, W has a strictly higher score than any other committee and the difference is $\frac{1}{d}$. This can be realized by the following ballot b_i. Consider $t-1$ categories, each representing a committee $W' \in \mathcal{P}_k(C) \setminus W$. For some $j \in [d]$, let B_i^j represent W'. Then $B_i^j = \bigcup_{c \in C \setminus W'}\{c^j\}$, i.e., the j-th category of ballot b_i is satisfied by any candidate that is not in W'. Hence, for every committee $W'' \in \mathcal{P}_k(C)$, there exists a candidate $c \in W''$ with $c^j \in B_i^j$ if and only if $W'' \neq W'$. Latter especially holds for W, yielding a score of $\frac{t-1}{d}$. We can extend b_i, by setting up more blocks of $t-1$ attribute categories. For example, if the score of a committee W should be $\frac{z}{d}$ higher than that of any other committee $W' \neq W$, it is sufficient to set up $z \cdot (t-1)$ categories. Accordingly, we can set up the score of any other committee in relation to all other committees. By simulating cardinal values for each committee (using relative scores due to a lack of normalization), we can easily induce any weak ranking $\succsim = \succsim_i$. □

GEMÄSS §5 DER PROMOTIONSORDNUNG VOM 15.06.2018.

EIDESSTATTLICHE ERKLÄRUNG

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 Konferenzberichten veröffentlicht, als Zeitschriftenartikel publiziert, oder bei Workshops vorgestellt:

"Irresolute Approval-based Budgeting"
↪ Baumeister, Boes, and Seeger [25]
↪ Eigenanteil: Siehe Abschnitt 4.4 auf Seite 49.

"Complexity of Manipulative Interference in Participatory Budgeting"
↪ Baumeister, Boes, and Hillebrand [22]
↪ Eigenanteil: Siehe Abschnitt 5.4 auf Seite 57.

"Time-Constrained Participatory Budgeting Under Uncertain Project Costs"
↪ Baumeister, Boes, and Laußmann [24]
↪ Eigenanteil: Siehe Abschnitt 6.4 auf Seite 77.

"Complexity of Sequential Rules in Judgment Aggregation"
↪ Baumeister, Boes, and Weishaupt [13, 14]
↪ Eigenanteil: Siehe Abschnitt 7.4 auf Seite 89.

"Collective Combinatorial Optimisation as Judgment Aggregation"
↪ Boes, Colley, Grandi, Lang, and Novaro [33, 34]
↪ Eigenanteil: Siehe Abschnitt 8.4 auf Seite 103.

"Distortion in Attribute Approval Committee Elections"
↪ Baumeister and Boes [21]
↪ Eigenanteil: Siehe Abschnitt 9.4 auf Seite 137.

"Bounded Approval Ballots: Balancing Expressiveness and Simplicity for Multiwinner Elections"
↪ Baumeister, Boes, Laußmann, and Rey [23]
↪ Eigenanteil: Siehe Abschnitt 10.4 auf Seite 145.

Ort, Datum

Linus Leopold Boes