Department of Informatics,
Athens University of Economics and Business

Algorithmic and Game-Theoretic Aspects of Computational Social Choice

Georgios Papasotiropoulos

ADVISOR: Evangelos Markakis (Associate Professor)

A dissertation submitted in partial fulfillment of the requirements for the degree of *Doctor of Philosophy*.

March'24

Approval for this dissertation has been granted by the thesis committee:

Evangelos Markakis (advisor)
Associate Professor, *Department of Informatics*, Athens University of Economics and Business.

Antonis Dimakis
Assistant Professor, *Department of Informatics*, Athens University of Economics and Business.

Aris Pagourtzis
Professor, *School of Electrical and Computer Engineering*, National Technical University of Athens.

Georgios Amanatidis
Senior Lecturer, *School of Mathematics, Statistics and Actuarial Science*, University of Essex.

Aris Filos-Ratsikas
Lecturer, *School of Informatics*, University of Edinburgh.

Dimitris Kavvadias
Assistant Professor, *Department of Mathematics*, University of Patras.

Alkmini Sgouritsa
Assistant Professor, *Department of Informatics*, Athens University of Economics and Business.

Abstract

The present dissertation aims to explore innovative decision-making approaches that complement traditional voting processes, examining them through an algorithmic, game-theoretic and axiomatic lens. The central objective is to identify voting procedures that can increase agents' desire to participate in collective governance and improve voters' participation experience. Therefore, we focus on suggesting and analysing voting frameworks and election rules that reconcile the varying preferences of the electorate towards achieving socially desirable outcomes in various scenarios, aspiring to elevate both the quantity and quality of community involvement in democratic processes.

The doctoral thesis addresses two primary challenges of Computational Social Choice:

➛ The first part concerns elections with a combinatorial structure, where a decision must be made over a set of interdependent issues of multiple alternatives each. In such scenarios, voters cast conditional approval ballots which enable them to express (approval) preferences for issues that are contingent on the outcome of others. We first focus on the winner determination problem under the natural voting rule that minimizes voters' total dissatisfaction, referred to as Conditional Minisum. We present positive and negative results for approximate and parameterized algorithms. Additionally, we investigate the robustness of the rule against the malicious actions of adding or deleting voters or alternatives, in terms of computational complexity. Finally, we introduce and study two further voting rules: Conditional Proportional Approval Voting and Conditional Method of Equal Shares. These are inferior to Minisum with respect to the total satisfaction score but, in contrast to Minisum, are able to ensure strong proportionality guarantees.

➛ In the second part, we focus on the concept of Delegative Voting, which strikes a balance between direct and representative democracy. Flexible and dynamic voting frameworks that empower voters to choose their preferred mode of participation are designed and analysed. First, we propose a framework that allows voters to express approval preferences not only for casting a ballot themselves or abstaining but also for being represented by specific sets of other voters. We then examine the problems of minimizing (resp. maximizing) the number of dissatisfied (resp. satisfied) voters from the perspectives of computational complexity, parameterized and approximate algorithms. Secondly, we study a delegative voting framework in which we incorporate a temporal dimension to address adaptations of voters' preferences over time, towards mitigating vote loss: a critical concern for Delegative Voting settings. Finally, we explore the potential enhancement of electorate's satisfaction through ballot delegation, in scenarios where voters have incomplete preferences and we establish necessary and sufficient conditions for achieving a socially better outcome –compared to direct voting– by leveraging the participation of proxies.

Περίληψη

Η παρούσα διδακτορική διατριβή αποσκοπεί στη διερεύνηση καινοτόμων προσεγγίσεων λήψης αποφάσεων που συμπληρώνουν τις παραδοσιακές διαδικασίες ψηφοφορίας, εξετάζοντάς τες από τη σκοπιά των αλγορίθμων, της θεωρίας παιγνίων και της αξιωματικής θεμελίωσης. Ο κεντρικός σκοπός είναι να προσδιοριστούν και να αναλυθούν εκλογικές διαδικασίες που μπορούν να αυξήσουν την επιθυμία των ατόμων για συμμετοχή στη συλλογική διακυβέρνηση και να βελτιώσουν την εμπειρία της συμμετοχής των ψηφοφόρων. Στοχεύοντας σε κοινωνικά επιθυμητά αποτελέσματα για μια σειρά διαφορετικών σεναρίων, φιλοδοξούμε να βελτιώσουμε ποσοτικά και ποιοτικά τη συμμετοχή του κοινού στις δημοκρατικές διαδικασίες.

Η διατριβή επικεντρώνεται σε δύο βασικές προκλήσεις της Υπολογιστικής Θεωρίας Κοινωνικής Επιλογής:

- ➛ Το πρώτο μέρος επικεντρώνεται σε εκλογές με συνδυαστική δομή όπου απαιτείται η λήψη αποφάσεων για πολλαπλά αλληλοεξαρτώμενα ζητήματα για τα οποία οι ψηφοφόροι έχουν τη δυνατότητα να εκφράσουν προτιμήσεις εξαρτώμενες από την έκβαση άλλων ζητημάτων. Επικεντρωνόμαστε αρχικά στο πρόβλημα εύρεσης νικητή για κάθε ένα από τα ζητήματα, υπό τον φυσικό κανόνα ψηφοφορίας που ελαχιστοποιεί τη συνολική δυσαρέσκεια των ψηφοφόρων (Conditional Minisum). Παρουσιάζουμε θετικά και αρνητικά αποτελέσματα για προσεγγιστικούς και παραμετρικούς αλγορίθμους. Επιπλέον, διερευνούμε την ανθεκτικότητα του κανόνα έναντι των κακόβουλων ενεργειών προσθήκης ή διαγραφής ψηφοφόρων ή προσθήκης ή διαγραφής επιλογών σε ορισμένα ζητήματα, εξετάζοντας την υπολογιστική πολυπλοκότητα των αντίστοιχων προβλημάτων. Τέλος, εισάγουμε και μελετάμε τους κανόνες ψηφοφορίας Conditional Proportional Approval Voting και Conditional Method of Equal Shares. Αυτοί οι κανόνες είναι υποδεέστεροι του Minisum ως προς τη συνολική ικανοποίηση των ψηφοφόρων αλλά, σε αντίθεση με τον Minisum, εξασφαλίζουν ισχυρές εγγυήσεις αναλογικής εκπροσώπησης.
- ➛ Στο δεύτερο μέρος, εστιάζουμε στην έννοια του Delegative Voting, η οποία συνδυάζει χαρακτηριστικά άμεσης και αντιπροσωπευτικής δημοκρατίας. Σχεδιάζονται και αναλύονται ευέλικτα, δυναμικά μοντέλα εκλογών, τα οποία δίνουν στους ψηφοφόρους τη δυνατότητα να επιλέξουν τον προτιμώμενο τρόπο συμμετοχής τους στη διαδικασία. Αρχικά, προτείνουμε ένα πλαίσιο που επιτρέπει στους ψηφοφόρους να εκφράζουν τις προτιμήσεις τους όχι μόνο σχετικά με την καταχώρηση ψήφου από τους ίδιους αλλά και σχετικά με την εκπροσώπηση από ορισμένα σύνολα άλλων ψηφοφόρων. Έπειτα αναλύουμε τα προβλήματα ελαχιστοποίησης (αντίστοιχα μεγιστοποίησης) του αριθμού των δυσαρεστημένων (αντίστοιχα ικανοποιημένων) ψηφοφόρων από την οπτική της υπολογιστικής πολυπλοκότητας, των παραμετρικών και των προσεγγιστικών αλγορίθμων. Επίσης, μελετάμε ένα πλαίσιο ψηφοφορίας το οποίο ενσωματώνει τη χρονική διάσταση για να αντιμετωπιστούν προσαρμογές των προτιμήσεων των ψηφοφόρων που πιθανώς συμβαίνουν κατά την πάροδο του χρόνου, με απώτερο σκοπό τη μείωση της απώλειας ψήφων: ένα κρίσιμο μειονέκτημα γνωστών μοντέλων Delegative Voting. Τέλος, εστιάζουμε στην ενίσχυση της ικανοποίησης των ψηφοφόρων μέσω ανάθεσης της ψήφου, σε σενάρια στα οποία οι ψηφοφόροι δεν έχουν σχηματίσει πλήρη γνώμη για όλα τα προς ψήφιση ζητήματα και προσδιορίζουμε αναγκαίες και ικανές συνθήκες για την επίτευξη ενός κοινωνικά ικανοποιητικού αποτελέσματος.

Opening Thoughts and Acknowledgments

A dissertation is often perceived as the primary achievement of a Ph.D. journey. Yet, I'm grateful for having realized that a Ph.D. holds much more depth. It's a blend of countless moments molding you both as a researcher and as an individual: from the elation of your inaugural paper acceptance, to the frustration of discovering an error in a submitted work; from the irritation at reviewers who failed to appreciate a work you invested months in, to the approval of your proposal for long-term funding or travel grant; from the gratifying feedback following engaging and successful oral presentations of yours in research environments, to numerous attempts –whether fruitful or not– to articulate your research interests to broad audiences; from joyful initial brainstorming and collaborations with great minds and friends, to discussions on where to submit and how to structure a recently completed work; from the thrill of stumbling upon an idea that feels like the missing piece in the puzzle you've been wrestling with for months, to the pleasurable downtime during research visits and conferences in places you hadn't envisioned traveling to; and the list extends far beyond. Such moments have culminated in a resounding realization: *I confidently affirm that opting for a Ph.D. ultimately yielded the finest and most enjoyable work I could have undertaken in these past years.*

But let us now shift our focus to the dissertation itself (yes, the one you're reading right now), which is basically an overview of a research effort that spans multiple years. While uncertain about the presence of groundbreaking ideas in the pages that follow, I can now with relief and satisfaction declare that it stands as a body of work *I'm immensely proud of.* Not only are the results presented in the ensuing chapters products of as hard and time-consuming work as I could have put in, but also every aspect of this thesis –be it a crucial theorem, a toy-example, a well-polished proof, an entry in the references, a challenging notation, or a minor sidenote[1]– has been meticulously selected and crafted with utmost care!

This chapter marks the end of my Ph.D. research. As I started writing it, I realized I've been dreaming about this moment for ages –doesn't every Ph.D. student dream about it? But if that's the case, why am I struggling to find the right words now that it's finally here? Where did all those thoughts disappear to? Surprisingly, it doesn't bring the overwhelming excitement I anticipated. I figured it out: surely, how could I be thrilled about concluding what has been a perfect period of life!? Once more: *a perfect period of life!* Oh, now feels like the right time to start giving credit to those who contributed in making it perfect.

The starting words of this part couldn't have been dedicated to anyone other than the one who has had the most significant academic impact on this journey, being the guiding force that helped me discern what I want to do for the rest of my life. To the one who weathered the storm of 2,265 emails from me (until 13/03/2024)

[1]: Speaking of sidenotes, have you ever find yourself annoyed, flipping to the end of a document to delve into the details of a reference, or frustrated by backtracking through pages to recall a definition, a proven lemma, or an earlier-stated assumption? Let's not even talk about those footnotes that can make you lose your train of thought, darting to the end of the page and then struggling to find your way back. Well, worry not! The 1.5 column format of this dissertation has your back (and, did you even observe how cutely the contents of each section appear?). Note that not every reference found its way to the margins –only the most crucial ones, lest the margins become overly cluttered and illegible– and yes, you may spot some ample, unused spaces in the margins on occasion –not the most aesthetically pleasing sight I must admit– but hey, compromises had to be made! Cheers to smooth reading!

and hasn't blocked my email address (at least, not yet). This is for the mentor who not only imparted invaluable lessons on research thinking but also equipped me with a survival kit for navigating the academic landscape. The person with whom I engaged in midnight research more times than I pulled all-nighters during my college days. The one who never pressured me to work on anything that didn't suit me and consistently offered freedom and support for my exploration of questions and techniques I found intriguing, while he really listened to my thoughts and preferences when it came to things like paper-writing, preparing presentations, and choosing venues. Plus his attention to my funding wasn't just about keeping me afloat, but also encouraging me to book flights to academic events taking place in locations I never thought I'd visit. From moments of career questioning, to the times I arrived at work without keys to my office: he was there for me! A ton of thanks to Vangelis: the best advisor I could have had.

Six wonderful people, each bringing their own flavor of inspiration to my academic journey, have graciously agreed to serve on my thesis committee. Antonis Dimakis and Aris Pagourtzis have been there for me since day one of my PhD, always ready to lend a hand with any paperwork and formalities needed for my studies (and I've definitely called on them a lot). Giorgos Amanatidis, Aris-Filos Ratsikas and Alkmini Sgouritsa have always been academic role models for me and I feel incredibly fortunate to have had the opportunity to collaborate with them, in one way or another. Giorgos and Aris have always been there to provide ideas, guidance, and support (and I'll make sure to keep all the lifesaving LaTeX secrets they shared with me close at hand). It's my ambition to one day match the trust, freedom and security that Alkmini (my work-life balance icon) provided us during our time as TAs for her Algorithms course. I am especially thankful to Dimitris Kavvadias for introducing me to the world of Algorithms and for showing me the way to academia since my undergraduate days. I'm truly grateful to each of them!

Immense gratitude to Piotr Skowron for extending an invitation to accompany him on the next phase of my research journey. His warm welcome and understanding of my constraints will always be deeply appreciated.

Thanks are also due to my coauthors Markus Brill, Jannik Peters, Philip Lazos, Giorgos Amanatidis, and Aris Filos-Ratsikas, who not only helped me learn a lot but also treated me as an equal. Additionally, a big thank you to Artem Tsikiridis for making our research such an entertaining and fun experience.

I will always feel beholden to Jérôme Lang and Piotr Skowron for their involvement in my academic visits. These experiences have been incredibly enriching, both personally and professionally. They've not only been enjoyable and productive but also inspiring, as I discovered how diverse research environments can be and connected with numerous exceptional researchers.

Extending my academic family, I want to give a shoutout to all my office-mates and colleagues in our research group, who made these early days in academia so memorable. Many thanks to everyone who made their way through the "Theory, Economics, and Systems Laboratory" of A.U.E.B., for our research discussions and, most importantly, the gossips and laughter. Special thanks to Georgios Stamoulis who, together with Vangelis, made every possible effort in creating such a lively and harmonious atmosphere in our lab, which, I will truly miss.

My gratitude also goes to Katia Papakonstantinopoulou, Nikos Leonardos, Giorgos Zois, and Antonis Dimakis for making my TAs an enjoyable experience.

Thanks to Paul Gölz for bringing the template of this thesis to my attention and for his guidance on typography.

Beyond academia, I've been lucky to have a circle of friends[2] who have always been there to offer, apart from encouragement, a much-needed distraction: while I may have occasionally tested their patience they've helped me see that a bit of socializing every now and then might not be entirely terrible.

My heartfelt thanks go out to my close family members first, and then to my extended family, for always having my back (and always trying to make sure that I'll never forget it); this journey wouldn't have been possible without their support. I owe you everything and hope to make you proud.

By the way, sorry to anyone I haven't gotten back to via calls or texts; PhD has been my handy excuse, but now it seems I'll need to come up with a fresh alibi. :)

But most of all, I'd like to express my deepest thanks to my happiness consultant, my caring guardian, my positivity guru, my constant motivation to become the best version of myself. Thanks for your unwavering respect and support. Thanks for boosting my mood when I need it the most. Thanks for brightening up our home every single day, and my life as a whole throughout all these years. Konstantina, thanks for being amazing!

Finally, I'd like to acknowledge the funding sources of my PhD studies. This research was primarily supported by the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the "3rd Call for H.F.R.I. PhD Fellowships" (Fellowship Number: 5163). Prior to receiving the scholarship, I have also received partial support from the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the "1st Call for HFRI Research Projects to support faculty members and researchers and the procurement of high-cost research equipment" (Project Number: HFRIFM17-3512) and from the General Secretariat for Research and Innovation under a grant to support the research activity of the "Theory, Economics, and Systems Laboratory" of A.U.E.B. I am grateful to the executive board of Archimedes Research Unit of Athena Research Center and the Research Department of Input Output Global for offering me positions in their research internship programs. Dissemination and research activities related to this dissertation have also received funding from the COST Action "European Network for Game Theory", the E.R.C. grant for the "Pro-Democratic" project, the Hellenic Artificial Intelligence Society, and the IJCAI-AIJ Travel Grant.

With such a lengthy paragraph of acknowledgments for funding, you might picture me penning my thesis amidst piles of cash. While I don't want to challenge that image, it's worth noting (or perhaps not?) that the reality of a PhD life in Greece normally involves a continuous pursuit of small, short-term grants. I won't dwell on it too much; I'm appreciative of my fortune, just looking forward to brighter days ahead.

[2] No one will judge me for forgetting to mention their name if I explicitly refer to none, right? ;)

Contents

FIRST WORDS

Introduction 1

The primary aim of this chapter is to outline the main topics and research questions addressed in the dissertation, along with highlighting the principal contributions of the associated research. Before delving into these aspects, we provide a brief exploration of the key facets of (Computational) Social Choice: a vast, yet relatively new, field.

The objective here is not to offer a comprehensive overview but rather to provide a concise (and, in parts, engaging) introduction, setting the stage for what follows. For those seeking an in-depth understanding of the field it is undeniable that there are plenty of thorough resources available; there is a multitude of captivating readings, each offering unique perspectives and insights into the realm of Computational Social Choice: references such as [ABE+19; ASS10; BCE+16; BCE12; CEL+07; End11; End17; FHH+09; HM15; LS23; Rot15] are just a starting point, and the list is far from exhaustive.[1]

A Road-Map of the Chapter.

In Section 1.1, we explore the pivotal role of preference aggregation in decision-making scenarios, spanning from mundane choices to complex digital landscapes, laying the foundation for Social Choice Theory.

Section 1.2 bridges Social Choice Theory and Computer Science in Computational Social Choice, which mainly tackles questions emerging from the former through the lens of the latter.

In Section 1.3 we discuss an illustrative example motivating Social Choice studies and innovative frameworks.

Section 1.4 envisions a democratic future by investigating frameworks for enhanced preference articulation and engagement while confronting the primary challenges of the thesis, discussing research directions, and formally defining and motivating its objectives.

In Section 1.5 we summarize the works that form the basis of this dissertation by offering an overview of the models, results, methods, and techniques that underpin the research.

[1] Some of these works introduced me to the field of Computational Social Choice, while others fueled my enthusiasm and helped shape the problems I explored during my Ph.D.

1.1 Social Choice Theory: Navigating the World of Collective Decision-Making

In today's world, we are constantly exposed to a broad spectrum of dilemmas as we engage in decision-making. This could involve the choice between Indian, Thai, or Mexican cuisine for a group gathering; the municipal elections; selecting the ideal team-bonding exercise–be it an escape room challenge, a cooking class, an outdoor adventure or none of them–; the organization of the details for a work meeting; casting a yes-no vote in a referendum; the decision between a shared tube of weiss, lager, or tropical ale beer; choosing a book club, movie night, or board-game session for the next outing; contemplating a decision such as settling for a quiet suburban life or embracing the bustling cityscape for a lifelong choice within a family: these everyday situations underscore the pivotal role of preference aggregation. Naturally, these situations often find resolution through voting and election schemes as groups navigate the intricate web of preferences in their shared experiences, even if, sometimes, the voters do not explicitly recognize that they are, in fact, participating in an electorate.

We will now step outside routine decisions. Originally conceived to address everyday, political, and economic challenges, Social Choice Theory proves its versatility by expanding beyond its human-centric origins to include autonomous software agents. Consider your favorite web search engine, where webpage rankings can be turned into a complex decision-making process, and webpages, functioning as voters, express their preferences through interlinking. Extend this perspective to a meta search engine that seamlessly blends results from various search engines, treating each engine as a voter. The central challenge lies in aggregating these preferences into a decision that genuinely mirrors the collective will. Venturing further into the digital landscape, we encounter recommender systems—tools guiding users in product selection based on their preferences. This digital exploration extends also to the Semantic Web, where one can picture a scenario with disparate information providers presenting conflicting ontologies for the same concepts. Of course, the discussion could include far more examples, but these suffice to indicate that collective decisions are not only taken among humans, and, therefore, Social Choice finds wide applicability in multi-agent systems as well.

The field of Social Choice studies the aggregation (mapping) of individual preferences (input) towards a collective decision (output) by principally designing and evaluating theoretically a wide range of voting rules. It is primarily motivated by the obvious democratic premise that social policy and group choice should originate from the preferences of the society and it is heavily based on an axiomatic approach. Hence, in Social Choice Theory, a problem can be specified by a set of desired axioms and, in high level, it is considered as solvable, if one can find an aggregation procedure that meets the given criteria. At the same time, voting rules are also evaluated in terms of their susceptibility to manipulation, control, or other factors. This involves, among others, the misreport of preferences in order to enforce a more beneficial outcome or the attempt to control the outcome of the election, e.g. by the addition or deletion of voters, perhaps via bribing.

1.2 The Fusion of Computer Science and Social Choice: Computational Social Choice

Social Choice Theory does not consider computational aspects. As an example, in many settings, known winner determination procedures -even if they satisfy multiple well-desirable properties- bump into computational intractability (and thus are inapplicable in realistic scenarios). In addition, a voting rule might be theoretically susceptible to manipulation or control by malicious agents, but finding the right strategies for them may turn out to be a computationally hard problem (and thus such aspects may be less of a worry in reality). As a consequence, the viewpoint of Computer Science on many extents of Social Choice Theory, can have a significant contribution on the study of collective decision-making processes. Indeed, it led to the development of a promising research area, referred to as Computational Social Choice. The adaptable frameworks of Social Choice provide a mathematical lens through which one can explore the principles of collective decision-making across diverse domains. In the realm of Computational Social Choice, a dynamic interface between Computer Science and Economics unfolds, addressing the intricate design and analysis of methods for collective decision-making from the viewpoint of Computer Science.

Computational Social Choice (COMSOC) stands at the crossroads of Social Choice Theory and Computer Science, fostering a rich interchange of ideas between these two disciplines. Classic work in Social Choice Theory has traditionally concentrated on results regarding the existence of procedures meeting specific requirements, but the computational dimension has quite recently entered the spotlight. The interdisciplinary field of Computational Social Choice unfolds in dual dimensions. Firstly, it delves into the application of Computer Science techniques, such as complexity analysis and algorithm design, to scrutinize social choice mechanisms like voting procedures. In doing so, it brings computational perspectives into classic problems in Social Choice Theory. On the flip side, Computational Social Choice is equally concerned with importing concepts from Social Choice Theory into the realm of computing.

The landscape of Computational Social Choice is enriched by a fusion of ideas from various disciplines, including Theoretical Computer Science, Artificial Intelligence, Mathematical Logic, Political Science, and Economic Theory. In essence, Computational Social Choice emerges as a dynamic field bridging theoretical and computational aspects of collective decision-making. It grapples with the challenges of equitable representation in democracies, transcending the limitations of traditional voting systems. By infusing computational techniques and social choice concepts, the field pioneers the way forward, providing a solid theoretical foundation for designing inclusive and socially desirable decision-making procedures.

1.3 Everyday Voting Tales: A Gateway to Social Choice

A reader unfamiliar with Social Choice might have wondered while reading the previous sections: 'Elections? Voting? Isn't there a straightforward solution to all those settings? Do we really need a whole field for such studies? Why isn't it easy to decide on the outcome, if we are given the preferences of the voters?' The example that comes next serves to illustrate the answer to such inquiries.[2]

Imagine a scenario in which a (fictional?) character, is on the verge of completing his Ph.D, and his friends, to be called Voter 1, Voter 2 and Voter 3, are deliberating on the perfect surprise gift for this milestone.

One friend takes the lead, "*Alright, team, we're in a pickle. We need to choose a gift for the completion of his Ph.D. journey. Here are our options: an exciting cooperative board game, a large plush toy from a famous cartoon series, or an experience: we could organize a small conference on his behalf or even ask a comedian or a band to make him a private comedy or music show! What are your thoughts?*" Another friend, henceforth Voter 1, enthusiastically suggests, "*We should go for the unforgettable experience. Perfect for such a celebration*". Voter 2 counters, "*Wait! It is a fact that he had enough of such surprises in the last few years. I believe that the plush toy is the way to go. It's definitely his style, it characterizes his Ph.D. presentations and thus he'll love the idea.*" Another friend, Voter 3, chimes in, "*A board game has timeless charm. It's a classic choice, and as far as we know him, he will definitely appreciate it.*"

Now, the three friends who expressed their preferences face a trilemma: board game, plush toy, or experience? If there were only two choices, one of them should have more votes than the other, making the decision straightforward. But now, no two of the voters agree on the same option. So, they delve into a second round of discussions.

Voter 1 spoke up, "*I would definitely vote for the experience. Regarding the rest, I don't have a strong opinion, but for the shake of completeness, I put the plush toy second in my preference order, and lastly, the board game.*" Voter 2 shared, "*I'm all about the toy first, followed by the board-game, and then the experience.*" Voter 3 added, "*The board game is my top choice, followed by the experience, and lastly, the toy.*"

	opinion
Voter 1	E>T>B
Voter 2	T>B>E
Voter 3	B>E>T

Table 1.1: Initial voters' preferences in the example discussed in Section 1.3. For a pair of alternatives a, b, we use $a > b$ to denote that a voter prefers option a to option b. We denote by B the board-game option, by T the plush-toy and by E the experience present.

[2] In fact, I had these concerns when I first delved into Social Choice Theory. Examples similar to the one that follows sparked my interest and motivated my studies.

As the friends pondered, one of them suggested, "*It looks like we now need to decide on a rule to determine the winner. How about each option goes head-to-head, and the one that beats every other option wins?*" The rest of the friends agreed, "*Given that most of us would prefer it over all the alternatives, it seems fair to conclude that this is the best option for us to purchase.*" As they voted, according to Table 1.1, a paradox unfolded: the experience beats the toy, the toy beats the board game, and the board game beats the experience! One of the friends exclaimed, "*Hold on! Did we just create a cycle of preferences? None of the three gifts won!*" Another laughed, "*Looks like choosing a gift for a surprise is full of surprises!*"

"*Let's not give up!*" Voter 3 said, "*You know, I don't have such strong preferences; I can change my vote to reach a decision. And I will go with the preferences of Voter 2; this should solve the problem, right?*". However, Voter 1 also decided to change his opinion, as he just realized how important the precise ranking of all the choices was. Initially, he thought that his favorite choice was the experience and didn't think much about the rest, so he put them in an arbitrary order. Now, he decided on the precise ranking of these options. Initially, he didn't have a strong preference between the toy and the board game, and now he decided to (still) go for the experience as his favorite choice but goes for the board game after it, leaving the plush toy as his least preferred option; "*A plush toy for a defense present? Ridiculous!*", he remarked.

But see what is happening now, as depicted in Table 1.2: two voters prefer the toy to the board game to the experience, and one prefers the experience to the board game to the toy. Voter 3 shouts out loud, "*We have a winner: we should buy the toy as, now, most of us are voting for it as our favorite option. We definitely have a winner! We can now eventually stop debating! Phew!*" Voter 1 mused, "*But, hold on, even though the toy is a choice that satisfies the majority, it is the worst option according to mine. On the other hand, the board-game isn't the worst for anyone. It's like a compromise choice that doesn't upset anyone too much and it is a safe present, don't you think?*" Voter 2 chimed in, "*Interesting point. So, there are two options, each with strong justifications for being the winner.*" He then got disappointed "*Meh, this will not lead anywhere!*"[3]

	opinion
Voter 1	E>B>T
Voter 2	T>B>E
Voter 3	T>B>E

Table 1.2: The modified voters' preferences, as discussed in the example of Section 1.3.

In this gift-giving conundrum, we lightly delved into the intricacies of decision-making. The takeaway from this scenario is that numerous, intuitive fairness criteria can give rise to diverse, yet natural and reasonable, voting rules which, in turn, can produce different rational outcomes, and in some cases, they might not even reach a conclusive decision. This inherent diversity in outcomes is one of the most intriguing aspects that captivates the study of Computational Social Choice. The exploration of such scenarios not only adds a layer of complexity to decision-making processes but also underscores the importance of designing rules that align with societal values and preferences.

[3] Should such an example be seen as an example that "*will not lead anywhere,*" as one of the participants said, or as an example that opens a whole world of voting systems? The Ph.D. studies of the main character of the discussed scenario might have made him lean towards the second. And you?

Before concluding this section, let’s see a few more hypothetical scenarios: Consider a setting where participants could buy any number of presents as a gift, and one insisted on purchasing the plush toy only if the board game accompanied it, while another voter agreed to buy any set of presents but not all of them due to budget constraints. Also, imagine a voter who keenly observed that another participant possessed superior insights into what would genuinely make the main character of this section happy and would like to delegate all her voting power directly to that voter. Although these scenarios may seem more intricate to implement and analyze, they carry significant meaning, offering voters the opportunity to express nuanced preferences. As we progress to the following sections and chapters, it will become evident that such preferences constitute the focal point of the current dissertation.

1.4 Towards a More Perfect Democracy: Innovations in Decision-Making Frameworks

Picture a world where election systems offer truly equal rights, and are transparent, and enjoyable for all. In this envisioned future, innovative frameworks can pave the way by revolutionizing how we articulate preferences, guaranteeing fair representation, encouraging robust citizen engagement, and adapting to the dynamic demands of our continually evolving communities.

Democracy has long been a cornerstone of society, but its implementations have been far from perfect. Although democracies often assert that every voter has an equal voice, the fact is that equitable representation is not always properly achieved, often hampered by varying interpretations of the notion of fairness. Furthermore, what if everyone not only had an equitable voice in the decision-making process, but also was able to express her true preferences? Expressing true preferences in voting is not always a straightforward matter, despite how deceptively simple it may appear at first. Cognitive burdens, e.g. information overload, can hinder the ability of the voters to fully express their preferences, while voting framework’s restrictions can further constrain their choices. Owing to these concerns, the experience of the participants in traditional election platforms is frequently suboptimal and uninspiring, leaving them disenchanted or even disillusioned with the process. Voting participation experience is a fundamental aspect arising when we think about the ideal democracy. Here is where Computational Social Choice comes into play to provide solid principles for designing and evaluating election procedures that produce desirable outcomes, paving the way for more engaging and inclusive democratic processes.

The current thesis aims to investigate, from a computational and game-theoretic perspective, novel decision-making approaches that complement conventional voting systems, while aligning with today’s needs and preferences, with a focus on decision support systems for multi-agent decentralized environments. These systems are widely used in various applications where agents can correspond to human or other entities such as robots, software agents or computer servers and they have preferences over elements, which must be aggregated into a collective decision. Scientific domains like Public Policy, Information and Communication Technologies, Recommender Systems, IoT, E-commerce and E-governance, Transportation/ Healthcare/ Education Management, rely on collective decision-making processes and efficient voting frameworks, algorithms, and techniques. Thus, the investigation of new decision-making approaches that improve the experience of the agents in preference aggregations is of utmost importance.

By providing an engaging and efficient voting experience that reconciles differing preferences of strategic entities, the goal is to contribute to the enhancement of democratic systems, fostering a sense of active participation and ownership in the democratic process among the electorate. The main objective is to design efficient, fair, and engaging decision-making techniques for multi-agent environments, that align with the complex needs and combinatorial preferences of modern societies.

As we navigate the complex and multifaceted challenges of the 21st century, the role of digital technologies in decision-making processes has become increasingly important as we are facing unprecedented challenges that require innovative solutions. Fortunately, recent research has shown that the field of Computational Social Choice offers promising avenues for addressing these challenges. To further advance the field, this thesis focuses on two research directions. An aspect that has posed several research challenges concerns voting over combinatorial domains, and a second one that has breathed new air in the field is the concept of delegative voting. The effects of these captivating features on the design of computationally and socially efficient decision-making processes is the focus of the dissertation.

Starting with the first feature, it is undeniable that nowadays the study of decision-making should imperatively examine problems over multi-issue domains with a combinatorial structure (i.e., in an election over multiple interdependent issues or in a committee-selection problem). For illustration purposes, consider a group of co-authors having to decide the date of a meeting, and whether it will be by physical or virtual participation. A voter who has an opinion about the date that is strongly conditioned on the meeting form (due to availability constraints for physical meetings), will not feel at ease voting independently for each issue. The same can hold in a committee election, where a voter prefers a certain candidate only if no other candidate with similar qualifications is elected. Allowing an increased level of expressiveness, by letting the voters exhibit preferential dependencies between their alternatives, can cause a complexity blowup and, thus, non-trivial challenges appear. In practice, there exist already various platforms for decision-making and winner-determination[4], however, they cannot handle successfully scenarios with logically interdependent alternatives. To conclude, the research objectives that emerge along this front are:

[4] Indicatively, we refer to some that have been implemented by researchers from the Computational Social Choice field: Whale (from Grenoble Informatics Laboratory), Vodle (from Potsdam Institute for Climate Impact Research), Stable Voting (see [HP23]) PrefTools (see [LRK23]), RoboVote (from Carnegie Mellon and Harvard), PB Stanford (from Stanford's University Crowdsourced Democracy Team), Opra (see [CQW+21]), Pnyx (from Technical University of Munich).

Objective 1: Investigate ways to achieve adequate tradeoffs between expressiveness and computational efficiency.

Objective 2: Explore the strategic implications introduced by voting in combinatorial domains.

The second feature of interest is motivated by the presence of natural obstacles in various types of elections (electorates are often inadmissibly large, keeping abreast of many different subjects so that to form an opinion often comes at a high price, high cost of communicating agents' preferences, etc). Hence, direct democracy is considered unimplementable on several occasions. At the same time, the traditional format of representative democracy has received criticism for allowing the voters to express their opinion only at predetermined times and without the option to revoke their support under any circumstances (one-off voting may lead to the effect of a misuse of authority when the winner of an election deviates with no consequences from campaign promises). Such considerations have led to more flexible models of participation in decision-making that have recently enriched the agenda of Computational Social Choice. Under the paradigm of delegative voting, voters are allowed either to vote themselves or to delegate their vote to another (purportedly more informed) voter, and withdraw or change their authorization at any time if they feel they are not being represented correctly. Thus, election formats that try to achieve a compromise between direct and representative democracy have appeared and could be seen either as an algorithmic problem with various optimization metrics or as a game-theoretic problem where strategic players attempt to maximize their own utility. The main objectives along this forefront are:

Objective 3: Design mechanisms indicating appropriate voters' delegations, towards socially desirable outcomes.

Objective 4: Examine strategic aspects in delegation procedures, when viewed as multi-player games.

1.5 Overview of Thesis Contribution

The present dissertation aims to explore innovative decision-making approaches that complement traditional voting processes, looking at them through algorithmic and game-theoretic perspectives. The main goal is to design and analyze voting procedures that can increase the desire of the agents to engage in collective governance and enhance the overall participation experience. The study looks at effective techniques for agregating the preferences of the participants towards socially desirable outcomes across various scenarios. In essence, the dissertation establishes certain guarantees and advocates for the adoption of voting procedures that significantly upgrade voters' expressiveness. The examined frameworks have the potential to elevate both the quantity and quality of community involvement in democratic processes.

The dissertation is expected to make a significant impact and contribute substantially to both practical applications and theoretical advancements. On the practical side, the research aims to enhance the user experience in elections, potentially leading to higher and finer civic participation. The algorithmic findings may have further effects on multi-agent systems, distributed computing, and information retrieval, particularly in areas like resource sharing, recommendation systems, and coordination in large-scale systems. In terms of theoretical contributions, the dissertation seeks to advance the state of the art in voting within combinatorial settings and delegative voting. The research examines social choice rules that are appealing, axiomatically righteous, and computationally tractable. More precisely, this dissertation is divided into two parts, each addressing a primary challenge within the domain of Computational Social Choice:

➛ The first challenge (refer to Section 1.5.1 for more details) concerns elections with a combinatorial structure, exploring decisions over interdependent issues where the voters cast conditional ballots. These ballots allow the voters to express preferences for an issue that are contingent on the outcome of other issues. The study first focuses on the winner determination problem under the natural voting rule that minimizes the voters' total dissatisfaction (referred to as Conditional Minisum), presenting tractability and intractability results for approximate and parameterized algorithms. Additionally, the robustness of the rule against the malicious actions of adding or deleting voters or alternatives, is investigated in terms of computational complexity. Finally, we introduce and study two further voting rules which (in contrast to Minisum) ensure strong guarantees in terms of proportional representation, yet a classic desirable property in Social Choice. Those rules are inferior to Minisum with respect to the total satisfaction score and form generalizations of well-studied rules into the setting of conditional ballots. Given that the examined framework represents a natural and more expressive generalization of the classic approval voting setting (a prevalent method that not only has been extensively studied in theory but also employed in numerous elections globally) our findings consistently support its application in real-world scenarios. From this thesis, a compelling suggestion emerges: the transition from classic approval systems to their conditional counterparts can significantly enhance the experience of the voters.

➛ In the second challenge (see Section 1.5.2 for more details) we focus on the concept of delegative voting, which strikes a balance between direct and representative democracy. Flexible, dynamic voting frameworks that empower the voters to choose their preferred mode of participation are designed and analysed. First, we study a framework that allows the voters to express approval preferences regarding representation by certain sets of other voters, self-casting ballots, or abstention and we analyse the problems of minimizing (resp. maximizing) the number of dissatisfied (resp. satisfied) voters from the perspectives of computational complexity, parameterized and approximate algorithms. Secondly, we study (from a viewpoint that lies in the middle ground between algorithmic and axiomatic approaches) a delegative voting framework in which we incorporate a temporal dimension to address adaptations of the preferences of the voters over time towards mitigating vote loss: a critical concern for such settings. Finally, the potential enhancement of voters' satisfaction through ballot delegation in the presence of incomplete preferences is investigated, identifying necessary and sufficient conditions for achieving a socially better outcome by leveraging the participation of proxies. Delegative democracy has found extensive applications in recent times, making its mark within political parties, corporate structures, and civic engagement initiatives in regional governments. As the body of theoretical studies that supports its application continues to grow, we foresee an expanding array of real-world applications and use cases in the times ahead.

In the following, we give an overview of methods and techniques that are most commonly used in this thesis.

Algorithmic Approaches: To implement the proposed methodologies effectively, it is crucial that the optimal solutions of the examined rules (or approximations of them with provable guarantees) can be efficiently computed, particularly in certain well-motivated scenarios. Consequently, a substantial portion of this thesis is dedicated to developing polynomial time procedures that can optimally or approximately determine the output under various collective choice rules possessing desirable properties.

Computational Complexity Analysis: In scenarios where providing a polynomial time algorithm for a specific collective choice rule or problem seems unattainable, for completeness, our objective is to present evidence suggesting the unlikelihood of existence of such an algorithm. In these cases, we offer reductions from well-established hard problems, illustrating that, unless widely accepted computational complexity assumptions fail, no polynomial time algorithm exists for solving the given problem. In certain situations, we also employ analogous techniques to demonstrate that there are no significantly superior algorithms compared to the proposed ones, either in terms of running time or approximation guarantees. Interestingly, in contrast to the typical applications of computational complexity, some problems addressed in this thesis view a hardness result as advantageous. For instance, consider the scenario discussed in Sections 1.1 and 1.2 where malicious agents attempt to influence election outcomes through nefarious actions. There, showcasing the computational difficulty of the manipulation problem holds value.

Axiomatic Analysis: A classic approach in Social Choice Theory involves applying the axiomatic method to delineate the properties of proposed voting rules, towards justifying their use in real-life situations. Essentially, given a decision-making problem and several desirable properties (referred to as axioms), the question arises: does a social choice rule satisfying all axioms exist? In cases where such a rule cannot be found, can we demonstrate the incompatibility of a specific set of axioms?

While the majority of our work draws on algorithmic concepts and techniques and computational complexity findings to address challenges arising from Social Choice Theory, as demonstrated in this thesis, delving into the realm of democratic innovations proves to be equally beneficial for Computer Science. This investigation not only introduces novel theoretical questions and techniques that can be thoroughly studied independently of a Social Choice motivation but also provides opportunities for substantial practical impact on the field.

We now focus on the specifics of the key contributions of each chapter. As mentioned, the thesis is organized into two parts: the first investigates conditional approval voting, while the second explores models of delegative voting.

1.5.1 Conditional Approval Voting

Chapters 2 and 3 are based on joint works with Evangelos Markakis [MP20; MP21b]. In these we focus on a generalization of the classic Minisum approval voting rule, introduced by Barrot and Lang (2016), and referred to as Conditional Minisum (CMS), for multi-issue elections with preferential dependencies. Under this rule, the voters are allowed to declare dependencies between different issues, but the price we have to pay for this higher level of expressiveness is that we end up with a computationally hard rule. Motivated by this, in Chapter 2 we focus on finding special cases that admit efficient algorithms for CMS. Our main result in this direction is that we identify the condition of bounded treewidth (of an appropriate graph, emerging from the provided ballots) as the necessary and sufficient condition for exact polynomial algorithms, under common complexity assumptions. We then move to the design of approximation algorithms. For the (still hard) case of binary issues, we identify restrictions, under which we provide the first multiplicative approximation algorithms for the problem. The restrictions involve upper bounds on the number of dependencies an issue can have on the others and on the number of approved alternatives per issue in a voter's ballot. In Chapter 3, we investigate the complexity of problems related to the strategic control of conditional approval elections by adding or deleting either voters or alternatives and we show that in most variants of these problems, CMS is computationally resistant against control. Overall, we conclude that CMS can be viewed as a solution with a satisfactory tradeoff between expressiveness and computational efficiency, when we have a limited number of dependencies among issues, while at the same time exhibiting sufficient resistance to control.

[MP20] Markakis and Papasotiropoulos (2020): Computational Aspects of Conditional Minisum Approval Voting in Elections with Interdependent Issues.

[MP21b] Markakis and Papasotiropoulos (2021): Winner Determination and Strategic Control in Conditional Approval Voting.

Chapter 4 is based on joint work with Markus Brill, Evangelos Markakis and Jannik Peters [BMP+23]. In this, we again consider the multi-issue election setting over a set of possibly interdependent issues that has been described in the previous paragraph, this time with the goal of achieving proportional representation of the views of the electorate. To this end, we employ a proportionality criterion suggested recently in the literature, that guarantees fair representation for all groups of voters of sufficient size. For this criterion, there exist rules that perform well in the case where all the issues have a binary domain and are independent of each other. In particular, this has been shown for Proportional Approval Voting (PAV) and for the Method of Equal Shares (MES). In this paper, we go two steps further: we generalize these guarantees for issues with a non-binary domain, and, most importantly, we consider extensions to elections with dependencies among issues, where we identify restrictions that lead to analogous results. To achieve this, we define appropriate generalizations of PAV and MES to handle conditional ballots. In addition to proportionality considerations, we also examine the computational properties of the conditional version of MES. Our findings indicate that the conditional case poses additional challenges and differs significantly from the unconditional one, both in terms of proportionality guarantees and computational complexity.

[BMP+23] Brill et al. (2023): Proportionality Guarantees in Elections with Interdependent Issues.

1.5.2 Delegative Voting

Chapter 5 is based on a joint work with Evangelos Markakis [MP21a]. In this work, we study a Liquid Democracy framework where the voters can express preferences in an approval form, regarding being represented by a subset of the voters, casting a ballot themselves, or abstaining from the election. We examine, from a computational perspective, the problems of minimizing (resp. maximizing) the number of dissatisfied (resp. satisfied) voters. We first show that these problems are intractable even when each voter approves only a small subset of other voters. On the positive side, we establish constant factor approximation algorithms for that case, and exact algorithms under bounded treewidth of a convenient graph-theoretic representation, even when certain secondary objectives are also present. The results related to the treewidth are based on the powerful methodology of expressing graph properties via Monadic Second Order logic. We believe that this approach can turn out to be fruitful for other graph related questions that appear in Computational Social Choice.

[MP21a] Markakis and Papasotiropoulos (2021): An Approval-Based Model for Single-Step Liquid Democracy.

Chapter 6 is also based on a joint work with Evangelos Markakis [MP23]. As it has been evident until now, in recent years, the study of various models and questions related to Liquid Democracy has been of growing interest among the community of Computational Social Choice. A concern that has been raised, is that current academic literature focuses solely on static inputs, concealing a key characteristic of Liquid Democracy: the right for a voter to change her mind as time goes by, regarding her options of whether to vote herself or delegate her

[MP23] Markakis and Papasotiropoulos (2023): As Time Goes By: Adding a Temporal Dimension Towards Resolving Delegations in Liquid Democracy.

vote to other participants, till the final voting deadline. In real life, a period of extended deliberation preceding the election-day motivates the voters to adapt their behaviour over time, either based on observations of the remaining electorate or on information acquired for the topic at hand. By adding a temporal dimension to Liquid Democracy, such adaptations can increase the number of possible delegation paths and reduce the loss of votes due to delegation cycles or delegating paths towards abstaining agents, ultimately enhancing participation. Our work in [MP23], takes a first step to integrate a time horizon into decision-making problems in Liquid Democracy systems. Our approach, via a computational complexity analysis, exploits concepts and tools from temporal graph theory which turn out to be convenient for our framework.

Chapter 7 is based on joint work with Georgios Amanatidis, Aris Filos-Ratsikas, Philip Lazos and Evangelos Markakis [AFL+24]. We study elections where the voters are faced with the challenge of expressing preferences over an extreme number of issues under consideration. This is largely motivated by emerging blockchain governance systems, which include voters with different weights and a massive number of community generated proposals. In such scenarios, it is natural to expect that the voters will have incomplete preferences, as they may only be able to evaluate or be confident about a very small proportion of the alternatives. As a result, the election outcome may be significantly affected, leading to suboptimal decisions. Our central inquiry revolves around whether delegation of ballots to proxies possessing greater expertise or a more comprehensive understanding of the preferences of the voters can lead to outcomes with higher legitimacy and enhanced voters' satisfaction in elections where the voters submit incomplete preferences. To explore this, we introduce a model where potential proxies advertise their ballots over multiple issues, and each voter either delegates to a seemingly attractive proxy or casts a ballot directly. We identify necessary and sufficient conditions that could lead to a socially better outcome by leveraging the participation of proxies. Overall, our results enhance the understanding of the power of delegation towards improving election outcomes.

[AFL+24] Amanatidis et al. (2024): On the Potential and Limitations of Proxy Voting: Delegation with Incomplete Votes.

We conclude by noting that in order to maintain thematic coherence of the dissertation, a small part of the doctoral research conducted has been omitted from this dissertation. In particular, we have worked with Evangelos Markakis and Artem Tsikiridis [MPT22] on a problem not directly related to the voting theory concepts discussed so far. We studied a covering problem motivated by spatial models in crowdsourcing markets, where tasks are ordered according to some geographic or temporal criterion. Assuming that each participating bidder can provide a certain level of contribution for a subset of consecutive tasks, and that each task has a demand requirement, the goal is to find a set of bidders of minimum cost, who can meet all the demand constraints. Our focus was on truthful mechanisms with approximation guarantees against the optimal cost.

[MPT22] Markakis et al. (2022): On Improved Interval Cover Mechanisms for Crowdsourcing Markets.

Overview of Publications.

Concisely, the current dissertation is mainly based on the following *conference publications*, where in each of them, the author of the dissertation was the primary contributor or among the primary contributors.

1. Markakis Evangelos and Papasotiropoulos Georgios. *Computational Aspects of Conditional Minisum Approval Voting in Elections with Interdependent Issues.* In: Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI), 2020.

2. Markakis Evangelos and Papasotiropoulos Georgios. *Winner Determination and Strategic Control in Conditional Approval Voting.* In: Proceedings of the 30th International Joint Conference on Artificial Intelligence (IJCAI), 2021.

3. Markakis Evangelos and Papasotiropoulos Georgios. *An Approval-Based Model for Single-Step Liquid Democracy.* In: Proceedings of the 14th International Symposium on Algorithmic Game Theory (SAGT), 2021.

4. Brill Markus, Markakis Evangelos, Papasotiropoulos Georgios and Peters Jannik. *Proportionality Guarantees in Elections with Interdependent Issues.* In: Proceedings of the 32nd International Joint Conference on Artificial Intelligence (IJCAI), 2023.

5. Amanatidis Georgios, Filos-Ratsikas Aris, Lazos Philip, Markakis Evangelos and Papasotiropoulos Georgios. *On the Potential and Limitations of Proxy Voting: Delegation with Incomplete Votes.* In: Proceedings of the 23rd International Conference on Autonomous Agents and Multiagent Systems (AAMAS), 2024.

6. Markakis Evangelos and Papasotiropoulos Georgios. *As Time Goes By: Adding a Temporal Dimension Towards Resolving Delegations in Liquid Democracy.* Under Review.

An extended merging of the two publications that appeared first in the list resulted in the following *journal submission*:

1. Markakis Evangelos and Papasotiropoulos Georgios. *On the Complexity of Winner Determination and Strategic Control in Conditional Approval Voting.* Under Review.

The author of this thesis has also co-authored the following publication:

1. Markakis Evangelos, Papasotiropoulos Georgios and Tsikiridis Artem. *On Improved Interval Cover Mechanisms for Crowdsourcing Markets.* In: Proceedings of the International Symposium on Algorithmic Game Theory (SAGT). 2022

Part One

Conditional Approval Voting

Winner Determination in Conditional Approval Voting 2

Over the years, the field of Social Choice has focused more and more on decision making over combinatorial domains [LX16], which involves settings like *multi-winner elections*, e.g. for the formation of a committee, and elections for a set of issues that need to be decided upon simultaneously, often referred to as *multiple referenda*. At what follows, we focus on approval voting as a means for collective decision making on multiple issues with multiple alternatives each. Approval voting offers a simple and easy to use format for running such elections, by having voters express an approval or disapproval separately for the alternatives of each issue. There is already a range of voting rules that are based on approval ballots, including the classic Minisum solution, which for each issue selects the alternative with the highest support from the electorate, along with more recently introduced methods (outlined in the "Related Work" section).

[LX16] Lang and Xia (2016): Voting in Combinatorial Domains.

However, the rules most commonly studied for approval voting are applicable only when the issues under consideration are independent. As soon as the voters exhibit preferential dependencies between the issues, we have more challenges to handle. More precisely, voters' preferences on a specific issue may be conditioned upon the outcome of other issue(s) and this is not uncommon in practical scenarios: A resident of a municipality may wish to support public project A, only if public project B is also implemented; a faculty member may want to vote in favor of hiring a new colleague only if the other new hires have a different research expertise; a group of friends may want to go to a certain movie theater only if they decide to have dinner at a nearby location; festival organizers could choose to approve the inclusion of several musical acts in their lineup but decide to limit the number of acts to a small fixed number, e.g., due to budget constraints; a grant committee may approve funding for Project X, but only if Project Y didn't receive sufficient support from the committee members to be implemented. We can also consider another example with conditional preferences, taken from recommendation systems for online advertising: suppose an ad management service needs to make a personalized selection of ads, to be shown on Alice's favorite news website. For each slot (or area) in the advertising region of the site, there is a set of possible ads to choose from and the overall goal is to maximize the likelihood that Alice will click on one of these ads. Her likelihood to click depends on whether she encounters ads that strongly align with her interests. If we think of the slots as corresponding to issues, a recommendation could be made by looking at the data from users "similar" to Alice (voters), and their clicking behavior (approvals). Notably, these voters have conditional preferences, as their probability of clicking on an ad is influenced when a related ad appears in a nearby slot as the probability may increase for products frequently bought together or decrease when a product is defamed by the ad of another.

It is rather obvious that voting separately for each issue cannot provide a good solution in any of the above settings. Consequently, as detailed in the "Related Work" section, several approaches have been suggested to take into account preferential dependencies. Nevertheless, the majority of these works are suitable for rules where voters are required to express a ranking over the set of issues or have a numerical representation of their preferences instead of approval-based preferences. The first work that introduced a framework for expressing dependencies exclusively in the context of approval voting was by Barrot and Lang (2016). They defined the notion of a conditional approval ballot (where the voters can specify a dependency graph for the issues of the election in conjunction with their ballots) and introduced new voting rules, that generalized some of the known rules from the literature of the standard approval setting. Among the properties that were studied, it was also exhibited that, in general, a higher level of expressiveness implies higher computational complexity. More precisely, the Minisum solution (also frequently referred to as the Approval Voting rule) is known to be efficiently computable in the standard (unconditional) approval setting, but its generalization, referred to as *Conditional Minisum* (or CMS in short), was shown to be NP-hard. In the unconditional approval setting, the Minisum solution stands as the most straightforward method for selecting winning alternatives, and it has established itself as one of the primary election systems extensively examined in Economic Theory, Political Science and Computational Social Choice, being also widely used in practice [End13].

Given how central the Minisum solution is, and how practical conditional ballots can be in real-life scenarios, and in light of the computational challenges presented in [BL16], it becomes natural to investigate whether CMS admits exact algorithms for certain families of instances or approximation algorithms with provable guarantees. Progress on this front would allow us to draw conclusions on the applicability of approval-based elections in which voters are endowed with a significant degree of expressiveness.

[BL16] Barrot and Lang (2016): Conditional and Sequential Approval Voting on Combinatorial Domains.

Contribution.

We undertake a study of the Conditional Minisum voting rule, a.k.a. CMS, which attempts to minimize the total dissatisfaction score across all voters in conditional approval elections, from the viewpoint of algorithms and complexity. Our goal is to enhance the understanding on the complexity implications due to conditional voting for the winner determination problem under a rule that is known to be efficiently computable in the absence of dependencies between issues, with respect to both exact and approximate solutions.

In Section 2.2, we focus on conditions that lead to exact polynomial time algorithms for computing the optimal solution under CMS voting rule. For this, we consider the intuitively simple (but still NP-hard) case, where each issue can depend on at most one other issue for every voter, and our main insight is that one can draw conclusions by looking at (undirected variant of) the global dependency graph of an instance, which is formed by taking the union of dependencies by all voters. We later generalize this result for dependencies on any constant number of issues. Restrictions on the structure of the global dependency graph allow us to identify the condition of bounded treewidth as the only restriction that leads to optimal efficient algorithms. More precisely, our results provide characterizations for the families of CMS instances that can be placed in P and FPT, implying that the condition of bounded treewidth serves as the lynchpin between expressiveness of voters' ballots and efficiency of solving the winner determination problem. These results also establish a connection with well studied classes of Constraint Satisfaction Problems, which can be of independent interest.

In Section 2.3, we provide the first multiplicative approximation algorithms for conditional approval elections of issues with binary domain, under the condition that for every voter, each issue can depend on at most one other issue. The considered family of instances, includes the set of instances that were proven to be NP-hard in [BL16]. In the corresponding graph-theoretic representation of the problem, which will be introduced in Section 2.1, the condition corresponds to voters with dependency graph of maximum in-degree no more than 1. The main positive result of the section is an algorithm that achieves an approximation factor of 1.1037. Interestingly, our algorithm is based on a reduction to MIN SAT, an optimization version of SAT that has rarely been applied in Computational Social Choice (in contrast to MAX SAT). The result is contingent upon an additional, but well-motivated from the perspective of Social Choice, assumption regarding the number of approved alternatives per issue in a voter's ballot. Imposing such a further requirement might at first seem demanding, however, we have also established a strong negative result: in the absence of further assumptions, no algorithm can attain any bounded multiplicative approximation guarantee, even for instances with binary domains and even if for every voter, each issue depends on at most one other issue. Concluding the section, we put forth some additional (and similar in flavor) assumptions that, when satisfied, also enable the existence of provable approximation guarantees, albeit non-constant. Interestingly, these results also relate the considered voting rule with classic algorithmic problems.

Related Work.

Approval voting for multi-issue elections has gained great attention in the recent years, driven by its simplicity and practical potential. Apart from the classic Minisum solution [BF78; BF82; LS10; Web78], other rules have also been considered, such as the Minimax solution [BKS07], Satisfaction Approval Voting [BK15], families of rules based on Weighted Averaging Aggregation [ABL+15], Proportional Approval Voting and Chamberlin-Courant. The last two rules, as well as Minisum, can be captured by the general family of Thiele voting rules; for these (and other approval based rules) we refer to the very recently published book [LS23] and to the surveys [BF10; Kil10]. None of these rules however allow voters to express dependencies.

[BF78] Brams and Fishburn (1978): Approval Voting.
[BF82] Brams and Fishburn (1982): Deducing Preferences and Choices in the 1980 Presidential Election.
[LS10] Laslier and Sanver (2010): Handbook on Approval Voting.
[Web78] Weber (1978): Comparison of Public Choice Systems.

The first work that exclusively studied issues in approval-based elections is by Barrot and Lang (2016). Namely, three voting rules were proposed for incorporating such dependencies (including the Conditional Minisum rule that we consider here) and some of their properties were studied, mainly on the satisfiability of certain axioms. Conditional approval ballots have a clear resemblance with the well-studied model of CP-nets [BBD+04], which is a graphical representation of voters' preferences depicting conditional dependence and independence of preference statements under a ceteris paribus (all else being equal) interpretation, but, as it has been highlighted in [BL16], the two frameworks define different semantics and are incomparable.

[BBD+04] Boutilier et al. (2004): CP-Nets: A Tool for Representing and Reasoning with Conditional Ceteris Paribus Preference Statements.
[BL16] Barrot and Lang (2016): Conditional and Sequential Approval Voting on Combinatorial Domains.

Even if one moves away from approval-based elections, the presence of preferential dependencies remains a major challenge when voting over combinatorial domains. Several methodologies have been considered achieving various levels of trade-offs between expressiveness and efficient computation. Some representative examples include, among others, sequential voting [AEG+11; DPR+11; LX09; XC12], or completion principles for partial preferences [CL12; LL09]. Analogous attempts to increase the expressiveness of agents' ballots have been also examined in other subfields of Computational Social Choice; indicatively we refer to Participatory Budgeting [JST20; REH23], Judgement Aggregation [GE10] and Liquid Democracy [CGN21].

2.1 Election Framework and Definitions

Let $I = \{I_1, \dots, I_m\}$ be a set of m issues, where each issue I_j is associated with a finite domain D_j of alternatives. An *outcome* is an assignment of a value for every issue, and let $D = D_1 \times D_2 \times \dots \times D_m$ be the set of all possible outcomes. Let also $V = \{1, \dots, n\}$ be a group of n voters who have to decide on a common outcome from D.

Voting Format. To express dependencies among issues, we mostly follow the format described in [BL16]. Each voter $i \in [n]$ is associated with a directed graph $G_i = (I, E_i)$, called *dependency graph*, whose vertex set coincides with the set of issues. A directed edge (I_k, I_j) means that issue I_j is affected by I_k. We also let $N_i^-(I_j)$ be the (possibly empty) set of direct predecessors of issue I_j in G_i. We first explain briefly how the voters are expected to submit their preferences, before giving the formal definition. For an issue I_j that has no predecessors in G_i (in other words, its in-degree is 0), voter i is allowed to cast an unconditional approval ballot, stating the alternatives of D_j that are approved by her. In the case that issue I_j has a positive in-degree in G_i, then let $\{I_{j_1}, I_{j_2}, \dots, I_{j_k}\} \subseteq I$ be all its direct predecessors (also called in-neighbors). Voter i then needs to specify all the combinations that she approves in the form $\{t : r\}$ where $r \in D_j$, and $t \in D_{j_1} \times D_{j_2} \times \cdots \times D_{j_k}$. Every such combination $\{t : r\}$ signifies the satisfaction of voter i with respect to issue I_j in a given outcome, when that outcome contains all alternatives in t as well as the alternative r for the issue I_j. Both cases of zero and positive in-degree for an issue can be unified in the following definition of conditional approval ballots.

Definition 2.1 *A conditional approval ballot of a voter i over issues $I = \{I_1, \dots, I_m\}$ with domains $D_1, \dots, D_m$ respectively, is a pair*

$$B_i = \langle G_i, \{A_j, j \in [m]\} \rangle,$$

where G_i is the dependency graph of voter i, and for each issue I_j, A_j is a set of conditional approval statements in the form $\{t : r\}$ with $t \in \prod_{k \in N_i^-(I_j)} D_k$, and $r \in D_j$.

To simplify the presentation, when a voter has expressed a common dependency for $k > 1$ alternatives of an issue I_j, we can group them together and write $\{t : \{d_j^1, d_j^2, \dots, d_j^k\}\}$, instead of $\{t : d_j^1\}$, $\{t : d_j^2\}$, …, $\{t : d_j^k\}$. Additionally, for every issue I_j with in-degree 0 by some voter i, a vote in favor of d_j will be written simply as $\{d_j\}$, instead of $\{\emptyset : d_j\}$.

An important quantity for parameterizing families of instances is the maximum in-degree[1] of each graph G_i, namely $\Delta_i = \max_{j \in [m]}\{|N_i^-(I_j)|\}$. Let also $\Delta = \max_{i \in [n]} \Delta_i$. Given a voter i with conditional ballot B_i, we will denote by B_i^j the restriction of her ballot to issue I_j. Moreover, a *conditional approval voting profile* is given by a tuple $P = (I, D, V, B)$, where $B = (B_1, B_2, \dots, B_n)$.

Definition 2.2 *The global dependency graph of a set of voters is the undirected simple graph that emerges from ignoring the orientation of edges in the graph $(I, \bigcup_{i \in [n]} E_i)$, where E_i is the edge set of the dependency graph of voter i.*

[1] When Δ_i is large for some voter i, the input might become exponentially large. Alternatively, one could try a succinct way of representing ballots using propositional formulae. We will not examine further this issue, since for the cases that we consider, the in-degree is constant.

Example 2.1 As an illustration, we consider 3 co-authors of some joint research who have to decide on 3 issues:

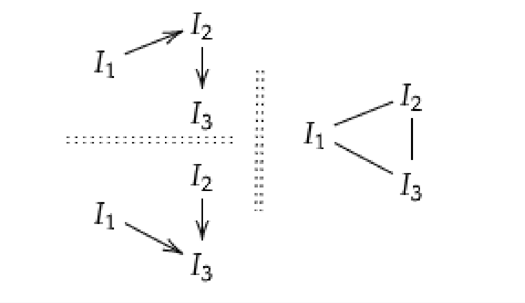
- Whether they will *work* more before the submission deadline on obtaining new theorems.
- Wether they have enough material to split their work into two, or even *multiple*, papers or submit all their results in a single submission.
- Whether they should invite a new *co-author* to work with them because of his insights that can help on improving their results.

The first author insists on more work before the submission, additionally he approves the choice of two submissions if and only if they work more on new theorems. Furthermore, he does not want to have a new co-author if and only if they split their work. The second author does not have time for more work before the deadline, he has no strong opinion on multiple submissions, approving both alternatives, and he agrees with inviting a new co-author only if they decide both to work more for new results and to submit a single paper. Finally, the last author is interested in working more and in splitting their work and she does not have a strong opinion on whether she prefers to invite a new co-author or not, unless they all decide not to work more neither to make more than a single submission, in which case she disagrees with such an invitation.

More formally, let $I = \{I_1, I_2, I_3\}$ be the aforementioned issues where $D_1 = \{w, \overline{w}\}$, $D_2 = \{m, \overline{m}\}$, $D_3 = \{c, \overline{c}\}$. Voters' preferences are shown below.

voter 1	voter 2	voter 3
w	$\{\overline{w}, m, \overline{m}\}$	$\{w, m\}$
$\overline{w} : \overline{m}$	$wm : \overline{c}$	$wm : \{c, \overline{c}\}$
$w : m$	$\overline{w}m : \overline{c}$	$\overline{w}m : \{c, \overline{c}\}$
$m : \overline{c}$	$w\overline{m} : c$	$w\overline{m} : \{c, \overline{c}\}$
$\overline{m} : c$	$\overline{wm} : \overline{c}$	$\overline{wm} : \overline{c}$

The dependency graphs of the instance follow. More precisely, in the figure below one can find the dependency graph of voter 1 (up left), the dependency graphs of voters 2 and 3 (down left) and the global dependency graph (right).



Voting Rule. In this work, we study a generalization of the classic Minisum solution in the context of conditional approval voting. To do so, we firstly define a measure for the dissatisfaction of a voter given an assignment of values to all the issues, using the following generalization of Hamming distance.

Definition 2.3 *Given an outcome $s = (s_1, s_2, \dots, s_m) \in D$, we say that voter i is dissatisfied (or disagrees) with issue I_j, if for the projection of s on $N_i^-(I_j)$, say t, it holds that $\{t : s_j\} \notin B_i^j$. We denote as $\delta_i(s)$ the total number of issues that dissatisfy voter i.*

Example 2.1 (cnt'd). The values of $\delta_i(s)$ for every outcome s and voter i follow.

$\delta_i(\cdot)$	wmc	$wm\bar{c}$	$w\overline{mc}$	$w\overline{mc}$	$\overline{w}mc$	$\overline{wm}\bar{c}$	$\overline{wm}c$	$\overline{wmc}$
voter 1	1	0	1	2	3	2	1	2
voter 2	2	1	1	2	1	0	1	0
voter 3	0	0	1	1	1	1	3	2

The rule that our work deals with is *Conditional Minisum* (CMS) and outputs the outcome that minimizes the total number of disagreements over all voters. To simplify notation, we will use CMS to refer both to the voting rule and to the related algorithmic problem; the exact meaning will always be clear from the context. Formally, the algorithmic problem that we study is as follows.

CONDITIONAL MINISUM (CMS)

Given: A voting profile P with m issues and n voters casting conditional approval ballots.

Output: A boolean assignment $s^* = (s_1^*, \dots, s_m^*)$ to all issues that achieves $\min_{s \in D} \sum_{i \in [n]} \delta_i(s)$.

Example 2.1 (cnt'd). The Conditional Minisum solution would prescribe to the authors to work more for new results, to split their work into two submissions, and not to invite a new co-author, which corresponds to the outcome $\{wm\bar{c}\}$.

If the global dependency graph of an instance is empty, i.e., $\Delta_i = 0$ for every voter i, then the election degenerates to Unconditional Minisum which is simply the classic Minisum rule in approval voting over multiple independent issues.

Finally, in the sequel, we will extensively make use of the *treewidth of a graph* G, denoted as $tw(G)$. For the relevant definition, we refer to [RS86] or to any textbook of parameterized complexity such as [CFK+15].

[RS86] Robertson and Seymour (1986): Graph Minors. II. Algorithmic Aspects of Tree-Width.

[CFK+15] Cygan et al. (2015): Parameterized Algorithms.

2.2 Optimal Algorithms

The price we pay for the higher expressiveness of CMS, compared to the classic Minisum solution, is its increased complexity. Here, we focus on understanding the properties that allow CMS to be implemented in polynomial time. For this, we firstly stick to the case where $\Delta_i \leq 1$ for every voter i, which is already NP-hard, and at the same time forms the most obvious, first-step generalization of Unconditional Minisum to the setting of dependencies. Then, we generalize our results for profiles of bounded Δ_i, for every voter i. To investigate what further restrictions can make the problem tractable, we utilize the global dependency graph of an instance, defined in Section 2.1, as the aggregation of all the dependencies of the voters into a single graph. To see how to exploit the global dependency graph, it is instructive to inspect the NP-hardness proof for CMS in [BL16], which holds for instances where $\Delta_i = 1$ for every voter i, and each dependency graph is acyclic. Examining the profiles created in that reduction, we notice that no restrictions can be stated for the form of the global dependency graph corresponding to the produced instances.[2]

[BL16] Barrot and Lang (2016): Conditional and Sequential Approval Voting on Combinatorial Domains.

Our insight is that it may not be only the structure of each voter's dependency graph that causes the problem's hardness, but in addition, the absence of any structural property on the global dependency graph. Motivated by this, we investigate conditions for the global dependency graph, that enable us to obtain the optimal solution in polynomial time. Our findings reveal that this is indeed feasible for the classes of graphs with constant treewidth.

In our results, we make extensive use of *Constraint Satisfaction Problems* (CSPs). A CSP instance is described by a tuple (V, D, C), where V is the set of variables, D is the Cartesian product of the domains of the variables, and C is a set of constraints. Each constraint involves a subset of the variables, and is represented by all the combinations of variables that make it satisfied. We will pay particular attention to the so-called *binary* CSPs, where each constraint involves at most two variables. The decision problem for a CSP asks whether we can find an assignment to the variables of V so that all constraints of C are satisfied, whereas a natural optimization version [FW92] is to minimize the number of unsatisfied constraints. When analyzing CSPs, a useful concept in the literature is the *primal* (or *Gaifman*) graph of an instance, defined below.

Definition 2.4 *The primal graph of a CSP instance is an undirected graph, whose vertices are the variables of the instance and there is an edge between two vertices, if and only if they co-appear in at least one constraint.*

The proof of the following theorem is based on formulating our problem as minimizing the number of unsatisfied constraints in an appropriate binary CSP instance, whose primal graph has constant treewidth. For these classes of CSPs, one can use known results from [Fre90] or [KHK02] for solving them efficiently.

[Fre90] Freuder (1990): Complexity of k-Tree Structured Constraint Satisfaction Problems.

[KHK02] Koster et al. (2002): Solving Partial Constraint Satisfaction Problems with Tree Decomposition.

[2] This holds since, an acyclic dependency graph for every voter does not necessarily lead to an acyclic global dependency graph and furthermore, the bounded in-degree in each G_i, does not imply a constant upper bound for the maximum in-degree of the global graph.

Theorem 2.1 *If the global dependency graph of a* CMS *instance with $\Delta_i \leq 1$, for every voter i, has constant treewidth, then* CMS *can be implemented in polynomial time, even with an arbitrary domain cardinality for each issue.*

Proof. Consider an instance $P = (I, D, V, B)$ of CMS with n voters and m issues, and let G be its global dependency graph. Suppose the treewidth of G is bounded by $k \in O(1)$, and let d be the maximum cardinality among the domains. We form an instance of the minimization version of binary CSP, with $m \cdot n$ constraints, where each constraint expresses the satisfaction of a specific voter for a specific issue.

Recall that we have assumed the maximum in-degree of every voter's dependency graph is at most one, thus each constraint in the CSP instance that we construct involves at most two variables, which means that the obtained CSP is indeed binary. Also, we can express each constraint by providing at most d^2 combinations of the two involved variables (i.e., the combinations that satisfy the constraint). Hence, the construction of the CSP instance can be done in polynomial time.

Since each constraint that involves two variables[3] corresponds to an edge of the global dependency graph and constraints with exactly one variable do not contribute any edges neither to the primal nor to the global dependency graph, the following can be easily verified.

Claim 2.2 *The primal graph of the produced* CSP *instance is identical to the global dependency graph of the* CMS *instance.*

Therefore, CMS has been formulated as minimizing the number of unsatisfied constraints in a binary CSP with primal graph of constant treewidth and these classes of CSPs are solvable in $O(n^k)$ time by [Fre90][4] or [KHK02]. □

We additionally highlight that the above theorem can be generalized when there is a weight w_i for each voter i so that the objective becomes the weighted sum of the dissatisfaction scores.

In trying to move away from treewidth-based assumptions, a natural question is whether we can solve other classes of instances, containing graphs of non-constant treewidth, by focusing on other parameters of the problem. Quite surprisingly, it turns out that bounded treewidth is essentially the only property that can yield efficiency guarantees. To establish this claim, we will first show a "reverse" direction to Theorem 2.1, namely that binary CSPs can be reduced to solving CMS. Hence, together with Theorem 2.1, this means that CMS is computationally equivalent to binary CSPs.

Binary CSPs, and therefore CMS as well, are also equivalent to a set of other problems such as the PARTITIONED SUBGRAPH ISOMORPHISM problem [Mar10].

[3] For uniformity, we could add dummy issues in the CMS instance (resp. dummy variables in the CSP instance) so that the final CSP only has constraints with exactly two variables.

[4] In fact, the original results in [Fre90] do not deal with the optimization version, but as demonstrated in later works (see e.g., Proposition 4.3 from [KKM+19], it can be extended for this version as well.

Theorem 2.3 *Every binary* CSP *with primal graph G, can be reduced in polynomial time to a* CMS *instance with $\Delta_i \leq 1$ for every voter i, and with G as the global dependency graph.*

Proof. For convenience, we will work with the standard decision version of CSP where one asks if there is a solution that satisfies all the constraints. Let P be a binary CSP instance, and without loss of generality, assume that every constraint involves exactly two variables (which can be enforced by the addition of dummy variables). We construct a CMS instance P', where the issues correspond to the variables and the voters correspond to the constraints of P. In particular, for every variable x_j of the CSP instance, we add an issue I_j and for every constraint we add a voter with the following preferences. Let x_j and x_k be the two variables involved in that voter's constraint.

- We pick one of the two variables (arbitrarily), say x_k, and we set I_k as the issue that the voter cares about, conditioned on I_j. We also set her conditional ballot for issue I_k in such a way, so that the voter becomes satisfied precisely for all combinations of values for x_j and x_k that make the constraint satisfied.
- The voter is satisfied unconditionally with every outcome of every issue other than I_j and I_k.

Obviously, in the described instance the dependency graph of every voter has maximum in-degree equal to one.

As an example, suppose that a constraint is of the form $x_1 \vee x_2$ and the variables x_1, x_2 have binary domain. Then we introduce a new voter, and two issues, I_1, I_2 (the issues may have been introduced already by other constraints in the instance), and we can select I_2 as being dependent on I_1. The conditional ballot regarding the satisfaction of the voter for I_2 is $\{x_1 : x_2\}$, $\{\overline{x_1} : x_2\}$, $\{x_1 : \overline{x_2}\}$. In addition, the voter has an unconditional ballot for I_1, in the form $\{x_1, \overline{x_1}\}$, thus approving every value for I_1.

To complete the reduction, we consider the decision version of CMS where we ask if there is an assignment with no dissatisfactions, i.e., the instance P' has an affirmative solution only when all voters are satisfied with all the issues. It is obvious that this is a polynomial time reduction (the conditional ballot of each voter for her single issue of interest can be described in $O(d^2)$ time, where d is the maximum domain cardinality of the CSP variables). It is quite obvious also that every edge from the primal graph of P corresponds to an edge in the global dependency graph of P', and vice versa. Hence:

Claim 2.4 *The primal graph of* CSP *instance P is identical to the global dependency graph of the* CMS *instance P'.*

Finally, it remains to see that there exists a solution to P' if and only if there exists a solution to P. Indeed, any solution to P' corresponds to an assignment of values to the issues such that all voters are satisfied with all issues, which means that all the constraints of the CSP instance P are satisfied. The converse is also easily verified. □

Theorem 2.3 allows us to apply some well known hardness results on binary CSPs, namely [Gro07; GSS01], which imply that one cannot hope to have an efficient algorithm for a class of CMS instances, if the class contains instances with non-constant treewidth. Hence, Theorem 2.1 is essentially tight, and this resolves the problem of finding a characterization for polynomial time solvability of CMS, subject to a standard complexity theory assumption. This is summarized in the following corollary.

[Gro07] Grohe (2007): Logic, Graphs, and Algorithms.
[GSS01] Grohe et al. (2001): When is the Evaluation of Conjunctive Queries Tractable?.

Theorem 2.1: If the global dependency graph of a CMS instance with $\Delta_i \leq 1$, for every voter i, has constant treewidth, then CMS can be implemented in polynomial time, even with an arbitrary domain cardinality for each issue.

Theorem 2.3: Every binary CSP with primal graph G, can be reduced in polynomial time to a CMS instance with $\Delta_i \leq 1$ for every voter i, and with G as the global dependency graph.

Corollary 2.5 *Let $\mathcal{G}$ be a recursively enumerable (e.g., decidable) class of graphs, and let CMS$(\mathcal{G})$ be the class of instances with a global dependency graph that belongs to $\mathcal{G}$, and with $\Delta_i \leq 1$ for every voter i. Assuming FPT $\neq$ W[1], there is a polynomial algorithm for CMS$(\mathcal{G})$ if and only if every graph in $\mathcal{G}$ has constant treewidth.*

Proof. The positive result comes from Theorem 2.1. By Theorem 2.3, an algorithm for the class of CMS instances whose global dependency graph belongs to $\mathcal{G}$ implies an algorithm for the CSP instances whose primal graph belongs to $\mathcal{G}$. The proof can be completed by applying the hardness results for binary CSPs by [Gro07; GSS01]. □

[Gro07] Grohe (2007): Logic, Graphs, and Algorithms.
[GSS01] Grohe et al. (2001): When is the Evaluation of Conjunctive Queries Tractable?.

Remark 2.1 If we strengthen the complexity assumption used in Corollary 2.5 to the Exponential Time Hypothesis (ETH), we can obtain an even stronger impossibility. In particular, by exploiting the result of [Mar10], and the proof of Theorem 2.3, we can show that under ETH, one cannot even hope for an algorithm on CMS$(\mathcal{G})$ that runs in time $f(G)||P||^{o(tw(G))/log(tw(G))}$, where $||P||$ is the size of the CMS instance and $G \in \mathcal{G}$. This implies that the running time $O(n^{tw(G)})$ of the algorithm from Theorem 2.1 is the best possible up to an $O(\log(tw(G)))$ factor in the exponent.

[Mar10] Marx (2010): Can You Beat Treewidth?.

2.2.1 Generalizations to Higher In-degrees

We highlight that Theorem 2.1 cannot be immediately generalized so as to apply to instances where $\Delta_i \geq 2$ for some voter i, since in that case the global dependency graph will not necessarily coincide with the primal graph of the corresponding CSP that we constructed in the proof of Theorem 2.1 (which is an essential part of the proof). In order to obtain a result for higher in-degrees, we introduce the following definition, which is a generalization of the global dependency graph, and where we simply replace a vertex and its in-neighbors by a clique on the same set of vertices.

Definition 2.5 *The extended global dependency graph of a set of voters is the undirected (simple) graph $(I, \bigcup_{i\in[n]} \tilde{E}_i)$, where $\tilde{E}_i$ is the edge set of a graph that corresponds to voter i and is created by enforcing an undirected clique for every issue I_j and any voter i, on the set $N_i^-(I_j) \cup \{I_j\}$.*

Note that for the cases where $\Delta \leq 1$, the extended global dependency graph of an instance coincides with the global dependency graph (and hence with the primal graph created in the proof of Theorem 2.1). The crucial observation now is that as long as $\Delta \in O(1)$, an instance of CSP equivalent to the initial CMS instance, can be created in polynomial time by following very closely the proof of Theorem 2.1. And most importantly, even though the extended global dependency graph of the CMS instance does not coincide with the global dependency graph, it does coincide with the primal graph of the created CSP instance; which is all we need. We stress also that one of the reasons we need $\Delta \in O(1)$, is to ensure that we can in polynomial time specify all the combinations that satisfy a constraint (namely by specifying at most d^{Δ} satisfying combinations).

To finalize the argument for the generalization, note that the created CSP instance will no longer be a binary CSP (i.e., it will not have at most two variables in each constraint). Nevertheless, these instances will have at most a constant number of variables in each constraint, due to Δ being constant, and they are still tractable as long as the primal graph of the CSP has bounded treewidth [Fre90]. Hence, our discussion can be summarized by the following theorem, which is indeed a generalization of Theorem 2.1 for instances of higher in-degrees.

[Fre90] Freuder (1990): Complexity of k-Tree Structured Constraint Satisfaction Problems.

Theorem 2.6 *If the extended global dependency graph of a CMS instance with $\Delta_i \in O(1)$ for every voter i, has constant treewidth, then CMS can be implemented in polynomial time, even with an arbitrary domain cardinality for each issue.*

Finally, we can also obtain a generalization of Theorem 2.3 (the exact same arguments apply with the global dependency graph being replaced by the extended global dependency graph). This leads to the following characterization regarding instances with higher in-degrees, which is the analog of Corollary 2.5.

Corollary 2.7 *Let $\mathcal{G}$ be a recursively enumerable class of graphs, and let CMS$(\mathcal{G})$ be the class of instances with an extended global dependency graph that belongs to $\mathcal{G}$, and with $\Delta_i \in O(1)$ for every voter i. Assuming FPT $\neq$ W[1], there is a polynomial algorithm for CMS$(\mathcal{G})$ if and only if every graph in $\mathcal{G}$ has constant treewidth.*

We finally note that a remark analogous to Remark 2.1 also applies here, but again for the treewidth of the extended global dependency graph.

2.2.2 Parameterized Complexity of CMS

The algorithm used in the proof of Theorem 2.1, runs in time exponential in $tw(G)$, where G is the global dependency graph and thus it places CMS in the complexity class XP, with respect to the treewidth parameter. One can wonder if anything more can be said concerning the fixed parameter tractability of the problem. Given the equivalence of our problem for $\Delta_i \leq 1$, for every voter i, with binary CSP, we can use existing results [GS08; SS10] to extract some further characterizations and obtain an almost complete picture with respect to the most relevant parameters. On the positive side, we can see that our problem is in FPT with respect to the parameter "treewidth + domain size". On the negative side, we cannot hope to prove FPT only with respect to the one of the two parameters, independent of the other, as stated below.

[GS08] Gottlob and Szeider (2008): Fixed-Parameter Algorithms for Artificial Intelligence, Constraint Satisfaction and Database Problems.
[SS10] Samer and Szeider (2010): Constraint Satisfaction with Bounded Treewidth Revisited.

Corollary 2.8 *When $\Delta_i \in O(1)$ for every voter i, CMS is in FPT with respect to the parameter $tw + d$, where tw is the treewidth of the extended global dependency graph and d is the maximum domain size. Moreover, even when $\Delta_i \leq 1$ for every voter i, it is $W[1]$-hard with respect to tw and with respect to d.*

Proof. First, let us introduce some notation for ease of presentation. Given a set of parameters S, we denote as $\Pi\{S\}$ the parameterized version of a problem Π, having all variables in S as parameters. $\Pi\{S\}$ is in FPT if every instance I of Π can be solved in time $O(f(S)|I|^c)$ for some constant c, and a computable function f, independent of any variable of Π other than the parameters in S. For a CSP instance we will denote by tw' the treewidth of its primal graph, by d' the maximum domain size of every variable, and by $arity$ the maximum number of variables that co-appear in a constraint.

To prove the positive statement, we exploit the fact that CSP$\{arity, d', tw'\}$ is in FPT by [GSS01]. This trivially implies that for CSP instances of constant arity, we have that CSP$\{d', tw'\}$ is in FPT. We can now use our Theorem 2.6. In particular, if we have a CMS instance, where $\Delta_i \in O(1)$, and where d is the maximum domain size and tw is the treewidth of the extended global dependency graph, Theorem 2.6 shows that we can reduce this to solving a CSP instance of constant arity and with $d' = d$ and $tw' = tw$. Hence, we have that CMS$\{tw, d\}$ is in FPT, when $\Delta_i \in O(1)$.

[GSS01] Grohe et al. (2001): When is the Evaluation of Conjunctive Queries Tractable?.

Theorem 2.6: If the extended global dependency graph of a CMS instance with $\Delta_i \in O(1)$ for every voter i, has constant treewidth, then CMS can be implemented in polynomial time, even with an arbitrary domain cardinality for each issue.

To prove the negative statements, we use the following definition: A set of parameters S dominates a set S' if whenever all parameters of S' are bounded by some constants, all parameters of S are bounded too. In [SS10] (Theorem 1 therein), it was proved that $\text{CSP}_{bin}\{arity, tw'\}$ and $\text{CSP}_{bin}\{arity, d'\}$ are W[1]-hard, where CSP_{bin} denotes the class of binary CSP instances. It is trivial to see that the set $S = \{tw'\}$ dominates the set $S' = \{arity, tw'\}$. Hence, by utilizing Lemma 1 in [SS10], we obtain that $\text{CSP}_{bin}\{tw'\}$ is $W[1]$-hard and the same is true also for $\text{CSP}_{bin}\{d'\}$. Given the reduction established in our Theorem 2.3, of binary CSPs with parameters tw' and d' to CMS instances with $\Delta_i \leq 1$ and with $tw = tw'$ and $d = d'$, we can conclude that both $\text{CMS}\{tw\}$ and $\text{CMS}\{d\}$ are W[1]-hard too. □

[SS10] Samer and Szeider (2010): Constraint Satisfaction with Bounded Treewidth Revisited.

On the Assumptions' Naturality.

We conclude by noting that the instances captured by the assumptions we have made in the current section are indeed meaningful in multi-issue elections with logically dependent issues. We mostly considered instances where $\Delta_i \leq 1$ for every voter i, which is the non-trivial (NP-hard to solve the winner determination problem) first-step generalization of the traditional (Minisum) approval voting rule. Secondly, the main positive result was for the case where the global dependency graph has a bounded treewidth. This can allow e.g., for paths, where we could think of the issues as being ordered on a line, with a sequential dependence between them. Likewise, when the global dependency graph forms a tree, we can again have a hierarchy regarding dependencies (e.g., a star graph can arise when there is a central issue, the decision for which influences the satisfaction of voters on the remaining issues). Going further, a constant treewidth allows for even more complex dependencies among issues, but still well-structured.

2.3 Approximation Algorithms

It is well known that a Minisum solution can be efficiently computed when there are no dependencies [BKS07]. In contrast to this, CMS is NP-hard even when all the issues have a binary domain and there is only a single dependence per voter, i.e., when every voter's dependency graph has just a single edge [BL16]. Given this hardness result, it is natural to resort to the framework of approximation algorithms. The only known result from this perspective is an algorithm by [BL16], with a *differential* approximation ratio of $4.34/(m\sum_{j\in I} 2^{|N^{-}(j)|} + 4.34)$, for the case of a common acyclic dependency graph, where $N^{-}(j)$ is the set of common in-neighbors of issue j (for each voter i and issue j, $N^{-}_i(j) = N^{-}(j)$). However, differential approximations (we refer to [DGP98] for the definition of this concept) form a less typical approach in the field of approximation algorithms. Instead, we focus on the more standard framework of *multiplicative* approximation algorithms, as treated also in common textbooks [Vaz03; WS11]. An

[BKS07] Brams et al. (2007): A Minimax Procedure for Electing Committees.

[BL16] Barrot and Lang (2016): Conditional and Sequential Approval Voting on Combinatorial Domains.

[DGP98] Demange et al. (1998): Differential Approximation Algorithms for Some Combinatorial Optimization Problems.

algorithm for a minimization problem achieves a multiplicative ratio of $\alpha \geq 1$, if for every instance I, it produces a solution with cost at most α times the optimal. We stress that a differential approximation ratio for minimization problems does not in general, imply any multiplicative approximation ratio [BP03].

[BP03] Bazgan and Paschos (2003): Differential Approximation for Optimal Satisfiability and Related Problems.

We start first with a rather strong negative result in terms of the viability of approximate solutions. The main result of the previous section was the hardness of computing optimal outcomes (Theorem 2.3). In fact, the proof of Theorem 2.3 implies also the following important multiplicative inapproximability.

Corollary 2.9 *Even when $\Delta_i \leq 1$ for every voter i, it is NP-hard to obtain any finite approximation ratio for* CMS.

Proof. If we look again at the proof of Theorem 2.3, we can see that we have reduced the solution of a binary CSP instance to deciding whether a CMS instance admits a solution of cost zero, i.e., a solution where all voters are satisfied. Given the hardness of binary CSPs, we conclude then that deciding if a CMS instance has optimal cost equal to zero is NP-hard. Suppose now that we could obtain an approximation algorithm with some finite approximation ratio for every instance. This immediately means that we could use this algorithm to distinguish between instances that have an optimal cost of zero (where the algorithm would have to return the optimal solution by the definition of approximation ratio) from the remaining instances (where the algorithm would return some solution with a positive cost). Hence we would have solved an NP-hard problem. □

Therefore, a polynomial time algorithm with a bounded multiplicative approximation guarantee, could only be possible under further assumptions. Our main contribution in this section is the first class of multiplicative approximation algorithms for some special cases of CMS. Sticking to the already hard class of binary domains and $\Delta_i \leq 1$, for every voter i (which includes the instances in which every voter has one edge, considered in the hardness result of [BL16]), we focus on instances that satisfy an assumption motivated by the fact that in the unconditional case, allowing voters to approve at most a single alternative per issue is already an interesting and well-studied voting scenario, which corresponds to the multi-issue analog of elections under the classic plurality setting.

Definition 2.6 *Consider a* CMS *instance with binary domains, and where the dependency graph of every voter i satisfies $\Delta_i \leq 1$. The instance is called* 1-*approval, if for every issue I_j that is dependent on some issue I_k according to the preferences of a voter i, it holds that i can be satisfied only with one pair, say $\{x_k : x_j\}$, with respect to I_j, where $x_k \in D_k$ and $x_j \in D_j$. No restrictions are imposed to the number of approved alternatives for unconditional ballots.*

To obtain a positive result, we first make use of known approximation algorithms for MIN k-SAT. Interestingly, minimization versions of SAT have rarely been applied in the context of Computational Social Choice, see e.g., [LMX18]. In fact it has hardly ever been used as a tool for obtaining approximation algorithms for other problems (we are only aware of an application for certain string comparison problems [GKZ05]). The use of MAX SAT is much more common, but for the case of CMS, and for multiplicative approximation guarantees, it does not seem convenient to exploit algorithms for maximisation problems. In a nutshell, if we use an approximation algorithm for MAX SAT, the conversion from the solution of a maximization problem to that of a minimization one that we have here, does not preserve a good approximation ratio for our objective function.[5] The main positive result of this section follows.

MIN k-SAT: A minimization version of SAT, where we are given a set of m clauses in k-CNF and we search for a boolean assignment so as to minimize the total number of satisfied clauses.

Theorem 2.10 *Let $\mathcal{F}$ be the family of* 1*-approval* CMS *instances, with binary domains and with $\Delta_i \leq 1$ for every voter i. Then any α-approximation algorithm for* MIN 2-SAT *yields an α-approximation algorithm for the family $\mathcal{F}$. In particular, we can have a polynomial time* 1.1037*-approximation for any* CMS *instance in $\mathcal{F}$.*

Proof. We present a reduction to MIN 2-SAT that preserves the approximation factor in the case where the given CMS instance is 1-approval. We first present a general reduction for any instance with $\Delta_i \leq 1$ for every voter i, which could be of broader interest. Later on, we will see how we can exploit this construction for 1-approval instances.

Consider an arbitrary instance P of n voters, with $\Delta_i \leq 1$ for every voter i, and let $I = \{I_1, \dots, I_m\}$ be the set of issues. We first create a logical formula C_{ij}, for every voter $i \in V$, and every issue $I_j \in I$, which indicates the cases where voter i is *not* satisfied with the outcome on I_j. For every issue I_j, recall that $D_j = \{d_j, \overline{d_j}\}$ is its domain, and x_j will be the corresponding boolean variable in the construction of C_{ij}.

For this we consider two cases. The first and easier is when for a voter i, and issue I_j, $N_i^-(I_j) = \emptyset$. All possible forms of B_i^j are depicted in the first row of Table 2.1, whereas the corresponding formula is shown in the second.

B_i^j	$\emptyset$	$\{d_j\}$	$\{\overline{d_j}\}$	$\{d_j, \overline{d_j}\}$
C_{ij}	$x_j \vee \overline{x}_j$	$\overline{x}_j$	x_j	$\emptyset$

Table 2.1: The formula when issue I_j has no predecessor in G_i.

On the other hand, if I_j has an in-neighbor (it can have only one by assumption), say $I_k \in I$, we set C_{ij} equal to the disjunction of all combinations of outcomes on issues I_j and I_k that dissatisfy voter i with respect to I_j.

To illustrate this construction, we describe an example with 4 voters, 2 issues $I = \{I_k, I_j\}$ and for every voter i, $G_i = \{I, \{I_k, I_j\}\}$. The preferences for

[5] The differential approximation result of [BL16] was based on the use of MAX SAT. But as stated earlier, this does not imply any non-trivial multiplicative approximation for CMS.

issue I_j are shown in the table below. Namely, for $i = 1, 2, 3, 4$, the first cell in the i-th row depicts B_i^j from which C_{ij} can be obtained as the disjunction of the ticked expressions in the remaining of the i-th row.

B_i^j	$(x_k \wedge x_j)$	$(x_k \wedge \overline{x}_j)$	$(\overline{x}_k \wedge x_j)$	$(\overline{x}_k \wedge \overline{x}_j)$
$\emptyset$	✓	✓	✓	✓
$\{d_k : d_j\}$		✓	✓	✓
$\{d_k : d_j\}$, $\{d_k : \overline{d_j}\}$			✓	✓
$\{d_k : d_j\}$, $\{\overline{d_k} : d_j\}$,$\{d_k : \overline{d_j}\}$				✓

Claim 2.11 *For an outcome $(s_1, \dots, s_m)$ of the issues and the corresponding assignment to the boolean variables $x_1, \dots, x_m$, voter i is dissatisfied with I_j if and only if the formula C_{ij} is true.*

The constructed formula C_{ij} is in DNF. To continue, we will need to make a conversion to CNF, which is easy to do given its small size as per the following lemma.

Lemma 2.12 *The formula C_{ij} for each voter $i \in V$, and each issue $I_j \in I$, can be written in CNF with at most 2 clauses, and where each clause contains at most 2 literals.*

Proof of Lemma 2.12. Fix a voter i and an issue I_j. For the cases where issue I_j has no in-neighbor in G_i, the lemma obviously holds, as can be verified in Table 2.1. For all other cases, I_j has a unique in-neighbor, say issue I_k, since we are dealing with instances where the in-degree is at most one. We now need to examine the form of C_{ij} for the cases that arise.

Case A. If voter i is satisfied only with 1 out of the 4 possible outcomes regarding I_j and I_k, then C_{ij} is a disjunction of 3 conjunctions. Let us assume that C_{ij} is in the form: $(x_j \wedge x_k) \vee (\overline{x}_j \wedge x_k) \vee (x_j \wedge \overline{x}_k)$. All other cases are handled in exactly the same way. The following equivalences can bring C_{ij} to the desirable form.

$$(x_j \wedge x_k) \vee (\overline{x}_j \wedge x_k) \vee (x_j \wedge \overline{x}_k) \equiv x_k \vee (x_j \wedge \overline{x}_k) \equiv$$
$$(x_k \vee x_j) \wedge (x_k \vee \overline{x}_k) \equiv x_k \vee x_j$$

Case B. If voter i is satisfied with 2 out of the 4 possible outcomes, then C_{ij} is a disjunction of 2 conjunctions. Without loss of generality, we can assume we have one of the following cases (all remaining cases can also be brought to one of these formats). As verified below, by the right hand side of each term, all cases can be brought into the desirable form.

$$\begin{aligned}
&(1.) \qquad (x_j \wedge \overline{x}_k) \vee (\overline{x}_j \wedge x_k) \equiv (x_j \vee x_k) \wedge (\overline{x}_j \vee \overline{x}_k) \\
&(2.) \qquad (x_j \wedge x_k) \vee (\overline{x}_j \wedge \overline{x}_k) \equiv (x_j \vee \overline{x}_k) \wedge (\overline{x}_j \vee x_k)
\end{aligned}$$

Case C. If voter i is satisfied with 3 out of the 4 possible outcomes regarding I_j and I_k, we take the conjunction expressing the outcome that causes dissatisfaction. E.g., $C_{ij} = x_k \wedge \overline{x}_j$, when i is satisfied with everything apart from $\{d_k : \overline{d_j}\}$. Thus, C_{ij} has 2 clauses with 1 literal each.[6]

Therefore, all formulas can be written in CNF with at most 2 clauses and where each clause contains at most 2 literals. ⊠

Using Lemma 2.12 to convert each C_{ij} to CNF, we can now create a MIN 2-SAT instance P' by the multiset[7] of all clauses appearing in the C_{ij}'s, i.e., appearing in the formula

$$C = \bigwedge_{i \in V, I_j \in I} C_{ij}. \tag{2.1}$$

We now try to exploit how the analysis so far can help us for 1-approval instances. We will first need to compare the optimal solution of the CMS instance with the optimal solution of the corresponding MIN 2-SAT instance.

Lemma 2.13 *Let P be a CMS instance and P' be its corresponding MIN 2-SAT instance produced as discussed in the proof of Theorem 2.10. Let also* $\texttt{opt}(P)$ *and* $\texttt{opt}(P')$ *be the values of the optimal solutions of the instances. If P is 1-approval, then it holds that* $\texttt{opt}(P') = \texttt{opt}(P)$.

Proof of Lemma 2.13. Consider an optimal solution of the CMS instance P. Every voter contributes to the cost of this solution precisely the number of issues with which she is dissatisfied. Consider now the corresponding MIN 2-SAT instance P', formed by the clauses of the constructed formula C from Equation (2.1). Let us look at the truth assignment to the variables of C, as dictated by the values of the issues in the optimal solution of P. We will provide an upper bound on the number of satisfied clauses of C.

Under the truth assignment, it holds that for every voter i and for every issue I_j for which i is dissatisfied with respect to I_j, the formula C_{ij} is true. By Lemma 2.12, any C_{ij} has at most two clauses which could be satisfied when C_{ij} is true. But in the case when P is 1-approval, then any fixed voter i either votes unconditionally on I_j or her ballot belongs to Case A from the proof of Lemma 2.12. In both cases, C_{ij} is formed by a single clause. Hence, by

[6] Typically, according to the definition of the conditional approval framework, a voter could also be satisfied with all 4 possible outcomes of I_j and I_k or disagree with all possible outcomes. However, one can consider such ballots as simple unconditional ballots, as no real dependence between the issues exists.

[7] Some clauses may happen to appear more than once in the final formula but there is no harm in keeping such duplicates.

looking at all the clauses of C that come from combinations (i, j), for which voter i is dissatisfied with respect to issue I_j, we get a number of satisfied clauses equal to $\texttt{opt}(P)$. Let us focus now on pairs (i, j), for which voter i is satisfied with respect to I_j. Then, the corresponding formula C_{ij} is false. If the ballot of voter i with respect to I_j is unconditional or if her ballot corresponds to Case A of Lemma 2.12, then C_{ij} does not have any true clauses. Therefore, for 1-approval instances, $\texttt{opt}(P')$, which is the total number of satisfied clauses of C under the selected assignment, equals $\texttt{opt}(P)$. ⊠

Our construction gives rise to the following algorithm for CMS, under the discussed assumptions:

```
Algorithm 1:                          ▷Input: 1-approval profile P
1: Create $P'$ from $P$ using Lemma 2.12 and Equation (2.1).
2: Run an $\alpha$-approximation of MIN 2-SAT on $P'$.
3: Set the value of $I_j$ in $P$ to the value of $x_j$ in $P'$.
```

To conclude the proof of Theorem 2.10, let $\texttt{sol}(P')$ be the cost of the solution to P' produced in step 2 of Algorithm 1, which equals the number of satisfied clauses in C by the truth assignment of the α-approximation algorithm. This corresponds to a solution for CMS and let $\texttt{sol}(P)$ be its total cost. We note that the total number of distinct pairs (i, j) for which voter i is dissatisfied by issue I_j can be no more than the number of the satisfied clauses of C, since each C_{ij} corresponds to a pair of a voter and an issue. Hence, together with Lemma 2.13, we have the following implications:

$$\texttt{sol}(P) \leq \texttt{sol}(P') \leq \alpha \cdot \texttt{opt}(P') = \alpha \cdot \texttt{opt}(P)$$

Thus, every α-approximation algorithm for MIN 2-SAT yields an α- approximation for CMS, as long as P is 1-approval. To obtain the claimed approximation ratio, we use the algorithm for MIN 2-SAT from [AZ05], which achieves an approximation factor of 1.1037. □

[AZ05] Avidor and Zwick (2005): Approximating MIN 2-SAT and MIN 3-SAT.

Although we have not been able to obtain a constant factor approximation for any other instance of CMS, the proof of Theorem 2.10 motivates the study of two more special cases of interest, for which we can obtain a positive result via different procedures. The central idea is that the general construction presented in Lemma 2.12 identifies 2 more cases, other than 1-approval, that may occur regarding the satisfaction of a voter with respect to. an issue. These are precisely the Cases B and C in the proof of Lemma 2.12. To define these two cases more formally, once again, we consider instances with binary domains, and with $\Delta_i \leq 1$ for every voter i. We will provide positive results for the families of instances that will be called OR-instances and XOR-instances. The former family could be seen as a generalized variant of antiplurality instances (a.k.a. veto, see

The proof of Theorem 2.10 also reveals why we cannot extend it to have a constant approximation for other than 1-approval instances. In particular, for instances that involve the Cases B and C, described in the proof of Lemma 2.12, we cannot guarantee that Lemma 2.13 will hold (all we need is that $\texttt{opt}(P') \leq \texttt{opt}(P)$, but this could be far from true).

e.g. [BCE+16] for more details); it contains instances in which every voter who casts a ballot for an issue I_j that is conditioned on the outcome of an issue I_k, approves exactly three out of the four possible combinations for these issues, or equivalently, is dissatisfied only with a single combination (the reason behind the name of this family will become clear when inspecting the proof of the approximate result that follows). The latter, includes instances in which for every issue I_j that is dependent on an issue I_k according to the preferences of a voter i, we assume that these issues are of a complementary nature, i.e., that voter i either wants I_j to be set to the same value as I_k or to the opposite (but not both). In other words, the satisfaction of the voter depends on the XOR value between I_j and I_k. In both families, we impose no restrictions for the issues that have no dependence on other issues.

Definition 2.7 *We say that a CMS instance where the issues are binary and the dependency graph of every voter i satisfies $\Delta_i \leq 1$ is an OR-instance if every voter who casts a ballot on an issue I_j that is conditioned on the outcome of an issue I_k is approving all but one combinations $\{x_k : x_j\}$, where $x_\ell \in \{d_\ell, \overline{d_\ell}\}$, for $\ell \in \{k, j\}$.*

Definition 2.8 *We say that a CMS instance where the issues are binary and the dependency graph of every voter i satisfies $\Delta_i \leq 1$ is a XOR-instance if every voter who casts a ballot on an issue I_j that is conditioned on the outcome of an issue I_k, is voting either for $\{d_k : d_j, \overline{d_k} : \overline{d_j}\}$ or for $\{d_k : \overline{d_j}, \overline{d_k} : d_j\}$.*

In contrast to the proof of Theorem 2.10, we are now going to reduce to and use algorithms for the MIN-2-CNF-DELETION problem for OR-instances and the MIN-UNCUT problem for XOR-instances. Likewise in [ACM+05], we are going to define these problems in a unified and convenient to us formulation, and we will consider them as special cases of the CONSTRAINT SATISFACTION PROBLEM (CSP) which also appeared in Section 2.2. Say that we are given a set of boolean variables $b_1, \ldots, b_n$ and a set of constraints C and the goal is to find an assignment that minimizes the number of unsatisfied constraints. The MIN-2-CNF-DELETION problem is the special case of CONSTRAINT SATISFACTION PROBLEM in which each constraint can be written in a 2-CNF form. More precisely we will focus on instances in which each constraint corresponds to a single clause in 2-CNF form, which has been called 2-CNF CLAUSE-DELETION problem in the literature [KPR+97]. The MIN-UNCUT problem is the special case of CONSTRAINT SATISFACTION PROBLEM in which each constraint is of the form $b_i \oplus b_j = 0$ or $b_i \oplus b_j = 1$ and has also been called 2-CNF$\equiv$ DELETION in the literature [GVY96].

[ACM+05] Agarwal et al. (2005): $O(\sqrt{logn})$ Approximation Algorithms for MIN UNCUT, MIN 2-CNF DELETION and DIRECTED CUT problems.

[KPR+97] Klein et al. (1997): Approximation Algorithms for STEINER and DIRECTED MULTICUTS.

[GVY96] Garg et al. (1996): Approximate Max-Flow Min-(Multi) Cut Theorems and their Applications.

Theorem 2.14 *Let $\mathcal{F}$ be the family of CMS instances where the issues are binary and the dependency graph of every voter i satisfies $\Delta_i \leq 1$. If $\mathcal{F}$ only contains XOR-instances (resp. OR-instances), then any α-approximation algo-*

rithm for MIN-UNCUT *(resp.* 2-CNF CLAUSE-DELETION*) yields an α-approximation algorithm for the family $\mathcal{F}$. In particular, we can have a polynomial time* $\log m$*-approximation for the class of XOR-instances and a polynomial time* $\log m \log\log m$*-approximation for the class of OR-instances.*

Proof. We start by proving the statement for XOR-instances. The result follows from an approximation preserving reduction of an instance $P \in \mathcal{F}$ to an instance P' of MIN-UNCUT. This reduction is similar to the one presented in the proof of Theorem 2.10 but, in this case, the satisfaction of a voter with respect to an issue should correspond to a satisfied constraint. Consider a voter i of P that has an unconditional ballot with respect to an issue I_j, say in favor of the alternative $x_j \in D_j$ (resp. $\overline{x_j} \in D_j$), then, her preference can be simply expressed as $x_j \oplus 0 = 1$ (resp. $x_j \oplus 1 = 1$). On the other hand, if the voter's ballot on I_j is conditioned on the outcome of I_k, due to the fact that P is a XOR-instance, her preferences can be expressed as $x_j \oplus x_k = 0$ or $x_j \oplus x_k = 1$. We have now created an instance P' of MIN-UNCUT, and, in analogy to the proof of Theorem 2.10, one can show that the costs of the optimal solutions of the two instances coincide. Similarly, it also holds that $\texttt{sol}(P) = \texttt{sol}(P')$, where $\texttt{sol}(P')$ corresponds to the cost of the solution of an α-approximation algorithm for MIN-UNCUT, whereas $\texttt{sol}(P)$ corresponds to the cost of the solution of the algorithm that transforms any XOR-instance P of CMS to an instance P' of MIN-UNCUT, as previously described, and then uses an α-approximation algorithm for MIN-UNCUT on P'. Utilizing the $\log n$ approximation from [GVY96] (Section 8 therein) for MIN-UNCUT, where n is the number of variables in the instance, we obtain a $\log m$ approximation for CMS, under the discussed assumptions.

When it comes to OR-instances, it suffices to observe that a voter casting a conditional ballot in such an instance could express his preferences with a logical formula of the following form: $(x_j \wedge x_k) \vee (\overline{x}_j \wedge x_k) \vee (x_j \wedge \overline{x}_k)$, for some $x_j \in \{d_j, \overline{d_j}\}$ and $x_k \in \{d_k, \overline{d_k}\}$. But such an expression can be equivalently written as $x_k \vee (x_j \wedge \overline{x}_k)$ which, in turn, is equivalent to $x_k \vee x_j$. This can be seen as a constraint that is formed by a single clause in 2-CNF. The rest of the arguments are identical to the case of XOR-instances and the approximation factor follows from [KPR+97] (Section 3.3 therein), where a $\log k \log\log k$ approximation algorithm is presented for 2-CNF CLAUSE-DELETION, for k being the number of variables in the formula. □

We note that slightly better approximation factors are possible for CMS, under both assumptions, leveraging results from [ACM+05]. However, this comes with the caveat of introducing randomization techniques, a debatable aspect in the context of Social Choice settings.

Concluding this section, we highlight the attainment of a bounded approximation guarantee for every conceivable scenario, for the cases where all voters that are casting conditional ballots approve either one, two, or three combinations of values. Therefore, we have achieved positive results across the spectrum, albeit exclusively under the assumption that voters are required to approve the same number of combinations per conditional issue.

2.4 Concluding Discussion and Future Directions

Our work is centered around the CMS rule, a relatively new and highly natural voting rule for expressing preferential dependencies with approval-based conditional ballots in elections over multiple interdependent issues. We focused on computational aspects of CMS elections, from the perspective of the winner determination problem using exact (polynomial and parameterized) and approximate algorithms. We conclude that CMS provides a satisfactory tradeoff between expressiveness and efficiency under certain assumptions. It is conceivable that approximation guarantees can be obtained for instances with higher expressiveness (i.e., higher in-degrees) than those considered in Section 2.3. Additionally, one can also consider other objective functions, such as the Conditional Minimax rule, defined also in [BL16], for which, algorithmic results remain elusive. In principle, one can take any other voting rule defined for approval ballots and explore potential generalizations in the setting with conditional approval ballots, as done for instance in [BMP+23] (refer to Chapter 4 for more details) with Proportional Approval Voting rule (PAV) and Method of Equal Shares (MES). Finally, as highly critical areas of future work, we emphasize the importance of obtaining real or synthetic data on elections over interdependent issues (currently nonexistent in public preference data libraries such as [MW13]) and of generating simulations that could complement our theoretical findings.

[BL16] Barrot and Lang (2016): Conditional and Sequential Approval Voting on Combinatorial Domains.

[BMP+23] Brill et al. (2023): Proportionality Guarantees in Elections with Interdependent Issues.

[MW13] Mattei and Walsh (2013): Preflib: A Library for Preferences http://www.preflib.org.

On the Complexity of Strategic Control in Conditional Approval Voting 3

Continuing the exploration of CMS rule, we now take a step towards examining potential threats, that could arise due to malicious behaviour within elections over interdependent issues. These explorations can, in principle, contribute to enhancing the fairness and transparency of elections, ensuring the overall integrity of the procedure, and paving the way for the development of algorithms aimed at detecting and preventing malicious attempts. In the realm of strategic considerations, the primary focal points within the Computational Social Choice literature revolve around questions of strategyproofness and election control. Strategyproofness is the axiom that is met when no voter can increase her satisfaction with respect to the rule's outcome by misreporting her true preferences; in contrast to the unconditional case, CMS is known to be manipulable [BL16]. In our work, we focus on elections' control and we study a spectrum of scenarios where the election conductor seeks to control the election outcome, so as to align with their own preferences, through various strategic actions. While, in many instances, a controller may be able to influence the input of the election so as to successfully enforce her will, we examine, from a computational complexity perspective, the question of whether the conductor can always and in polynomial time exert control over the outcome. To be more precise, we narrow our focus to worst-case scenarios, examining the existence of instances where the conductor encounters inherent computational challenges in achieving their desired outcomes. Similar questions form a very prominent research agenda within Computational Social Choice as it pertains to understanding the susceptibility of election systems.

Our findings that will be outlined below coupled with those from Chapter 2 indicate that CMS stands as a strong candidate for real-life applications as it strikes a favorable balance between voters' expressiveness and computational efficiency, all while exhibiting sufficient robustness against certain malicious efforts.

Contribution.

We continue the study of the Conditional Minisum voting rule, a.k.a. CMS, from the viewpoint of algorithms and complexity, this time with the objective of identifying whether the rule is immune to, or, at least, computationally resistant against malicious control by strategic actions. We initiate the algorithmic study of some standard notions of election control for CMS. The problems we examine concern the attempt by an external agent to enforce the election of certain alternative(s) in either one or every issue under consideration, by adding or deleting either voters (see Section 3.1) or alternatives (see Section 3.2). We consider a total

of 8 variants of this question, depending on the number of issues to be controlled and on whether we have addition or deletion of voters/alternatives. We provide a set of computational complexity results that give a complete picture with respect to the crucial parameters of the input in every one of the considered problems. Our findings reveal that CMS is sufficiently computationally resistant, against such moves.

In this chapter, we consider strategic aspects of CMS under the framework presented in Section 2.1 and study questions related to controlling an election of interdependent issues, which falls under the broad and well studied umbrella of influencing election outcomes. The versions of election control we consider lies within the standard approaches that have been used for studying the complexity of affecting election outcomes. For an extensive study on this topic, we refer to [FR16]. Indicatively, the study of such problems with adding or deleting voters or alternatives began with [BTT92] and some subsequent works are [FHH11; GRS+22; HHR07; LFZ+09; MPR+08; Yan19].

[FR16] Faliszewski and Rothe (2016): Control and Bribery in Voting

[BTT92] Bartholdi III et al. (1992): How Hard is it to Control an Election?

Suppose that there is an external agent (called *controller*) who has a strong preference for a specific value of some (or every) issue in a CMS election. One of the instruments for enforcing a desirable value for the issue(s) the controller cares about, is by enabling new voters to participate or by disabling some existing voters, which can be done for example by changing the criteria for eligibility of voters. Furthermore, a controller could add more choices for the issues under consideration or delete existing ones, towards enforcing her will. We refer to [CFN+17] for related examples and further motivation. Finally, it is reasonable to assume that the controller does not have unlimited power, and therefore, she is capable of adding or deleting only a certain number of voters or alternatives.

Each combination of control features (i.e., addition vs deletion, voters vs alternatives, single issue vs multiple issues) gives rise to a different control type, namely control either all or a single issue by deleting voters (CDV), by adding voters (CAV), by deleting alternatives (CDA), or by adding alternatives (CAA). In this manner, we obtain 8 distinct algorithmic problems. Following the terminology of [HHR07], we say that a voting rule is *vulnerable* to a certain control type, if the corresponding problem is always solvable in polynomial time. If the problem is $\mathcal{C}$-hard for a complexity class $\mathcal{C}$, we consider the rule to be *resistant* to the specific control type (typically $\mathcal{C}$ is the class NP). In the cases where it is not possible for a controller to affect the election towards fulfilling her will, independent of complexity theory assumptions, we say that the rule is *immune* to the corresponding control type. The formal definitions of the control problems appear in the following subsections and are adaptations to CMS elections, of the original definitions of control problems provided in [BTT92]. An overview of the results we obtained appear in Table 3.1.

Table 3.1: Results on Controlling CMS elections. R stands for (worst-case) Resistant (i.e. NP-hard), V for Vulnerable (i.e. polynomially solvable) and I for Immune (i.e. impossible, independent of complexity theory assumptions). For a CMS instance on n voters, we denote as Δ the maximum in-degree of every voter's dependency graph ($\Delta = \max_{i\in[n]} \Delta_i$) and d the maximum domain size.

	CDV & CAV			**CDA**			**CAA**		
	$\Delta = 0$	$\Delta = 0$	$\Delta = 1$	$\Delta = 0$	$\Delta = 1$	$\Delta = 1$	$\Delta = 0$	$\Delta = 1$	$\Delta = 2$
	$d = O(1)$	$d = \omega(1)$	$d = O(1)$	$d = \Omega(1)$	$d = O(1)$	$d = \Omega(n)$	$d = \Omega(1)$	$d = \Omega(n)$	$d = O(1)$
ALL	R	R	R	V	R	R	I	I	I
1	V	R	R	V	V	R	I	R	R

3.1 Controlling Voters

We start with the problems of adding or deleting voters for enforcing a specific outcome either for a single issue or for every issue of the election.

It's crucial to bear in mind that the notion of resistance is rooted in the realm of worst-case instances, supported by our NP-hardness results. These results indeed pose a barrier for controllers seeking to manipulate the outcomes, but this may not be true for every instance of the problem. In real-world scenarios, the susceptibility of CMS to such attempts may exhibit different behaviour.

Instance: A CMS election (I, D, V, B), where V is the set of registered voters, a set V' of yet unregistered voters with $V \cap V' = \emptyset$ (for use only by CAV), an integer quota q, a distinguished alternative $p_j \in D_j$ for a specific issue I_j or an outcome $p \in D$ (for the "ALL" versions) specifying an alternative for every issue.

Problem CAV-1 (resp. CDV-1): Does there exist a set $V'' \subseteq V'$ (resp. $V'' \subseteq V$), with $|V''| \leq q$, such that p_j is the value of issue I_j in every optimal CMS solution of the profile $(I, D, V \cup V'', B)$ (resp. of the profile $(I, D, V \setminus V'', B)$)?

Problem CAV-ALL (resp. CDV-ALL): Does there exist a set $V'' \subseteq V'$, (resp. $V'' \subseteq V$) with $|V''| \leq q$, such that p is the unique optimal CMS solution of the profile $(I, D, V \setminus V'', B)$ (resp. of the profile $(I, D, V \setminus V'', B)$)?

We now present our results for these 4 problems, exhibiting that it is not generally easy for a controller to enforce her will in such elections. In fact, computational hardness of controlling by adding or deleting voters can be established even for very simple forms of elections, without even the presence of conditional ballots, as shown in the two theorems that follow.

Remark 3.1 One has the option of either breaking ties in favor of the controller, if there are multiple optimal solutions in CMS (as in [DKN+11]), or demand that the controller's will is fulfilled in every optimal outcome. We focus on the second case, as is also done in the seminal paper of Bartholdi et al. [BTT92]. Additionally, it is possible that the controller has a strong opinion not just for a single or all issues, but for a subset of issues. As a starting point, we have chosen to consider the two extremes (and intuitively simpler versions).

Theorem 3.1 *CDV-ALL is NP-hard even for Unconditional Minisum and for a binary domain in each issue.*

Proof. To prove the NP-hardness, we will have a reduction from the VERTEX COVER problem. Thus we start with an instance $(G = (V, E), k)$, which asks if there is a vertex cover of size at most k, and create an instance P of CDV-ALL. For every edge $e \in E$, we add an issue I_e having two possible alternatives, and denote its domain by $D_e = \{d_e, \overline{d_e}\}$. For every vertex

$v \in V$, we add a voter voting unconditionally for d_e, if e is incident to v and being satisfied with both $\{d_e, \overline{d_e}\}$ otherwise. Let there also be 2 dummy voters who are satisfied only with $\overline{d_e}$ for every issue I_e. Hence, all the ballots are unconditional, and we have an empty global dependency graph. For the quota parameter, we use $q = k$, and suppose that the controller wants to enforce the alternative $\overline{d_e}$ for every issue I_e. This completes the description of the CDV-ALL instance, where the goal is to decide if there exists a set V'' of size at most q, such that deleting those voters enforces the controller's desirable outcome.

Suppose that there exists a vertex cover $S \subseteq V$ of G, of size at most k. Since each edge of G has at least one endpoint in S, by removing all voters that correspond to S, each alternative d_e loses at least one approval vote. Hence, d_e would cause two dissatisfactions to the dummy voters (the others are indifferent), whereas $\overline{d_e}$ causes at most one dissatisfaction. Therefore, selecting the alternative $\overline{d_e}$ for every issue I_e is the unique optimal solution,

For the reverse direction, suppose there exists a set of voters S, whose removal causes the outcome $(\overline{d_e})_{e \in E}$ to become the unique optimal solution. First, we may assume that S does not contain any of the dummy voters (otherwise, add them back to the instance, and the total dissatisfaction score will not be affected).

Suppose now that S is not a vertex cover in G, and that at least one edge e is not covered by S. But this means that the removal of S from the CDV-ALL instance will leave intact the two voters that are satisfied only with d_e, and therefore d_e can also be selected in an optimal solution (since it causes the same number of dissatisfactions as $\overline{d_e}$). Therefore, we have a contradiction to the fact that the removal of S resulted in the unique optimal solution with $\overline{d_e}$ selected for every issue I_e. □

Theorem 3.2 *CAV-ALL is NP-hard even for Unconditional Minisum and for a binary domain in each issue.*

Proof. The proof is a simple adaptation of a reduction given for almost the same problem but in the context of the classic (unconditional) approval voting rule in [HHR07]. We stress that we cannot directly establish NP-hardness by applying the result of that work because when there are no conditional ballots, the version of approval voting as defined there selects as winner(s) the candidates who have the highest number of approvals, whereas Unconditional Minisum selects only candidates who are approved by at least 50% of the voters. In the instances used in the reductions of [HHR07] (see Theorem 4.43 therein), there are losing candidates who are approved by more than 50% of the voters, hence their proofs do not apply directly.

[HHR07] Hemaspaandra et al. (2007): Anyone but Him: The Complexity of Precluding an Alternative.

We start with an instance P of EXACT-3-COVER (X3C) and we define a CMS election where the set of issues is $I = B \cup \{I_{m+1}\}$ and each issue has a binary domain, with $D_j = \{b_j, \overline{b_j}\}$ for $j \in [m]$, and $D_{m+1} = \{w, \overline{w}\}$. The set of voters is as follows:

- There are $k-2$ registered voters who are satisfied with b_j for $j \in [m]$, and with $\overline{w}$. They are dissatisfied with the complements of these alternatives.
- There is one registered voter who is satisfied only with $\overline{b_j}$ for $j \in [m]$ and with $\overline{w}$.
- There are n unregistered voters corresponding to the sets of X3C instance. The voter corresponding to S_i is satisfied only with the 3 alternatives of S_i, and with w.

In EXACT-3-COVER (X3C), we are given a universe $B = \{b_1, \ldots, b_m\}$ with $m = 3k$, and a collection of sets $\mathcal{F} = \{S_1, \ldots, S_n\}$ with $|S_i| = 3$, for every set S_i and the goal is to decide if there is an exact cover, i.e. a subcollection of sets from $\mathcal{F}$ such that each element of the universe belongs to exactly one of these sets.

To finish the description, we set the quota parameter q equal to k and the desirable outcome of the controller to be $(\overline{b_1}, \ldots, \overline{b_m}, w)$. Hence, the goal in the CAV-ALL instance is to decide if there exists a set of unregistered voters V'' with $|V''| \leq k$ such that adding V'' to the registered voters makes the desirable outcome the unique optimal solution.

Suppose now that there exists an exact cover in P. Since $m = 3k$, the cover consists of exactly k sets. Select as V'' the k unregistered voters corresponding to the cover. We now have a total of $2k - 1$ voters in the election. For the first m issues, the alternative b_j satisfies exactly $k - 1$ voters and dissatisfies k voters, hence the optimal solution selects $\overline{b_j}$ for $j \in [m]$. For the last issue, the value w satisfies k voters and dissatisfies the remaining $k - 1$ voters. Hence, the unique optimal solution when adding the set V'' is precisely $(\overline{b_1}, \ldots, \overline{b_m}, w)$.

For the opposite direction, suppose that there is a set V'' of unregistered voters, with $|V''| \leq k$, such that when adding them to the registered voters, the unique optimal solution is the controller's desirable outcome. First notice that this implies that $|V''| = k$, otherwise there is not enough support for w to be selected. The only other possibility would be to have $|V''| = k - 1$, but then we have a tie, and there would be more optimal solutions with $\overline{w}$ instead of w. Since for the other issues, each b_j already has a support by $k - 2$ registered voters, then none of them received a support by two or more of the added voters. But these voters express a support for a total of $3k = m$ such alternatives, therefore, each b_j for $j \in [m]$, receives support by exactly one of the added voters. □

The next step is to see whether the hardness results of Theorems 3.1 and 3.2 go through when the controller wishes to control just a single issue. For Unconditional Minisum this is not the case if we insist on a constant domain size for the designated issue. The reason is that this can be reduced to an FPT version of the well known SET MULTICOVER problem. The following is implied by [BFN+20].

[BFN+20] Bredereck et al. (2020): Mixed Integer Programming with Convex/Concave Constraints: Fixed-Parameter Tractability and Applications to Multicovering and Voting.

Proposition 3.3 *CAV-1 and CDV-1 can be solved in polynomial time for Unconditional Minisum if the domain size of each issue is constant.*

As a consequence, any potential hardness result for CAV-1 and CDV-1 would have to consider either non-constant domain sizes or conditional ballots. Indeed, we establish that either of these settings suffices to establish NP-hardness. We start with elections where at least one issue has a non-constant domain size.

Theorem 3.4 *CAV-1 and CDV-1 are NP-hard, even for Unconditional Minisum, but with non-constant domain size in at least one issue.*

Proof. We will only describe the proof of NP-hardness for CAV-1 and the same can be established for CDV-1 in a very similar fashion, using almost the same reduction.

We will have a reduction from the problem of controlling a classic approval voting election by adding voters, proved NP-hard in [HHR07]. We recall that in an approval voting election, voters express their approved set of candidates, and the winner (or winners in case of ties) is the candidate with the highest number of approvals. The control problem there is to ensure that a designated candidate is the unique winner of the election. Our reduction starts with an instance P of the control problem in approval voting, where V and V' are the registered and unregistered sets of voters respectively, p is a designated candidate, and q is a quota. The goal is to select a set $V'' \subseteq V'$ with $|V''| \leq q$, so that the approval voting rule, when run on the voters in $V \cup V''$ will select p as the unique winner.

We create an instance P' of CAV-1 where the sets of voters, registered and unregistered, are the same as in P. If the number of candidates in P is m, we create a single issue in P' whose domain has exactly m possible alternatives, and p is the designated alternative that the controller wants to promote in P'. For every voter in P (whether coming from V or V'), the corresponding voter in P' specifies an unconditional ballot on the single issue, containing only her approved options in P. We also use the same quota parameter q as in P. This completes the description of P', which can be clearly constructed in polynomial time.

It is now easy to see that there exists a set $V'' \subseteq V'$ of at most q voters so as to ensure that p will be the outcome on the single issue of P', using the CMS rule for the voters of $V \cup V''$, if and only if the same set of voters can ensure that p will be the unique winner in the approval voting election of P. Indeed, if the CMS rule, run on the voters of $V \cup V''$, selects the outcome p in the instance P', this means by the definition of the CMS rule that p causes the minimum number of dissatisfactions among all possible alternatives, i.e., it has the highest number of approvals. This directly yields that p will be the unique winner in the instance P. The reverse direction is easy to see as well, with the same reasoning. □

We now study the hardness of these problems when we have conditional ballots. As the next theorem shows, it suffices to consider only profiles where each issue may depend on at most one other issue.

Theorem 3.5 *CDV-1 and CAV-1 are NP-hard, when $\Delta \leq 1$, even for a binary domain in every issue.*

Proof. We will focus on proving the statement for CDV-1. The proof for CAV-1 is very similar, requiring minimal changes, that we discuss at the end. We will prove that CDV-1 with a binary domain for every issue, and with $\Delta = 1$ is NP-hard, using a reduction from the NP-hard version of CDV-1 with $\Delta = 0$ and non-constant domains (Theorem 3.4). We firstly remind the reader that the proof of Theorem 3.4 indicates that the problem CDV-1 with $\Delta = 0$ and non-constant domain sizes is NP-hard, even for the family of instances with just a single issue (that has a non-constant number m of different alternatives), where every voter casts only approval ballots for a subset of alternatives.

Our reduction starts from an instance P of CDV-1 on a single issue with non-constant domain size (obviously $\Delta = 0$, since we have only one issue), and creates an instance P' of CDV-1 with binary domain for every issue and with $\Delta = 1$. Say that the issue of P has the following m different alternatives: $\{d_1, d_2, \dots, d_m\}$. Then the instance P' consists of m different binary issues $I_1, I_2, \dots, I_m$ such that the alternatives of issue I_j are $\{d_j, \overline{d_j}\}$, for $j \in [m]$. Hence, the idea is that each alternative of the single issue of P now corresponds to a different issue in P' with positive and negative alternatives. Furthermore, if q was the quota in P, we will use the same quota in P'. Finally, if d_ℓ was the designated alternative in P, for some specific $\ell \leq m$, then we will have that d_ℓ is the designated alternative for issue I_ℓ in P'.

We describe now the set of voters in P' as well as their preferences. For every voter v of P, we add a voter v' in P', such that for any $j \in [m]$, if v was approving the alternative d_j for the single issue of P, then v' approves d_j concerning the issue I_j in P', otherwise, if v was not approving d_j in P, then v' is indifferent in P' and votes for $\{d_j, \overline{d_j}\}$. We also add a set of $m(m-1)L$ dummy voters, where $L = nm + q + 1$ and n is the number of voters in P. In particular, for every ordered pair of distinct alternatives of P, i.e., for every (k, j), with $k, j \in [m]$, and $k \neq j$, we include L voters voting $\overline{d_k}$ for issue I_k, $\{d_k : \overline{d_j}\}$ for issue I_j, and $\{d_t, \overline{d_t}\}$, for every other issue I_t, with $t \in [m] \setminus \{k, j\}$.

We will prove the following claim, enforced by the construction of P'.

Claim 3.6 *Let P'' be the conditional approval voting profile that is derived by the deletion of a set of at most q non-dummy voters from the instance P'. Then, in any optimal* CMS *solution of P'', there is exactly one issue I_j for which the selected alternative is d_j, and for every $k \neq j$, the selected alternative will be $\overline{d_k}$.*

Proof of Claim 3.6. Let us define first the set of outcomes where exactly one issue takes a positive value, i.e., let $\texttt{POS}_1 = \{\{y_1, y_2, \ldots, y_m\} : \exists i \in [m] : y_i = d_i, \forall j \in [m]\backslash\{i\} : y_j = \overline{d_j}\}$. We will firstly prove that for any solution that belongs to $\texttt{POS}_1$, the total dissatisfaction incurred by the set of dummy voters equals $m(m-1)L$. To prove that, we inspect an arbitrary outcome of $\texttt{POS}_1$, say $\{\overline{d_1}, \overline{d_2}, \ldots, \overline{d_{p-1}}, d_p, \overline{d_{p+1}}, \ldots, \overline{d_m}\}$, for some $p \in [m]$.

It is convenient to view the set of $m(m-1)L$ dummy voters of P'', as being partitioned in the following 3 sets:

- The $(m-1)(m-2)L$ dummy voters whose dependency graph consists of an edge (I_k, I_j), such that $k, j \neq p$. These voters are dissatisfied only with respect to issue I_j since they are voting for $\{d_k : \overline{d_j}\}$ and d_k is not selected. They are satisfied with respect to I_k since they are voting in favor of $\overline{d_k}$, which is elected, and they are indifferent (hence satisfied) with respect to all other issues.

- The $(m-1)L$ dummy voters whose dependency graph consists of an edge (I_p, I_j), such that $j \neq p$. These voters are dissatisfied only with respect to issue I_p since they are voting for $\overline{d_p}$ but d_p is elected. They are satisfied with respect to I_j since they are voting in favor of $\{d_p : \overline{d_j}\}$ and both d_p and $\overline{d_j}$ are elected. Finally they are satisfied with respect to any I_t, for $t \neq p, j$, since they are indifferent for these issues.

- The $(m-1)L$ dummy voters whose dependency graph consists of an edge (I_j, I_p) such that $j \neq p$. These voters are dissatisfied only with respect to issue I_p since they are voting for $\{d_j : \overline{d_p}\}$. Furthermore they are satisfied with respect to I_j since they are voting in favor of $\overline{d_j}$, and they are satisfied with respect to any other issue.

Hence, the total dissatisfaction score of any outcome in $\texttt{POS}_1$, due to the dummy voters is $m(m-1)L$. To count the dissatisfaction from the remaining non-dummy voters, we define the following quantity: let x_i be the number of voters in the original instance P (after deleting the voters that correspond to the ones deleted in P') who had d_i in their approval list in P. Then, it can be verified that the total dissatisfaction due to the non-dummy voters for the profile P'', is $\sum_{i\in[m]\backslash p} x_i$. Hence, to conclude, the total dissatisfaction score for P'', under any outcome that belongs to $\texttt{POS}_1$ is equal to $m(m-1)L + \sum_{i\in[m]\backslash p} x_i$.

We will now compare the dissatisfaction score of the outcomes in $\mathtt{POS}_1$ with all other possible outcomes. We can define analogously the set of outcomes $\mathtt{POS}_2 = \{(y_1, y_2, \dots, y_m) : \exists i, j \in [m] : y_i = d_i, y_j = d_j, \forall k \in [m] \setminus \{i, j\} : y_k = \overline{d_k}\}$. We will firstly prove that the total dissatisfaction incurred by dummy voters in outcomes from $\mathtt{POS}_2$ is at least $m(m-1)L + 2L$.

We inspect an arbitrary outcome of $\mathtt{POS}_2$, say $(y_1, \dots, y_m)$ with $y_p = d_p$, $y_r = d_r$, for some specific p, r, and also $y_j = \overline{d_j}$, for any $j \neq p, r$. We analyze the set of dummy voters, by partitioning them in 4 sets as follows:

- The $(m-2)(m-3)L$ dummy voters whose dependency graph has the edge (I_k, I_j), for some k, j where both $k, j \neq p, r$. These voters are dissatisfied only with respect to issue I_j since they are voting for $\{d_k : \overline{d_j}\}$ but $\overline{d_k}$ is elected. They are satisfied with respect to I_k since they are voting in favor of $\overline{d_k}$. Finally they are indifferent and hence satisfied with respect to all other issues.

- The $2(m-2)L$ dummy voters whose dependency graph has either the edge (I_p, I_j) or $(I_r, I_j))$, with $j \neq p, r$. We analyze the ones with the edge (I_p, I_j), and the same conclusion holds for the other case as well. These voters are dissatisfied only with respect to issue I_p since they are voting for $\overline{d_p}$, but d_p is elected. They are satisfied with respect to I_j since they are voting in favor of $\{d_p : \overline{d_j}\}$ and both d_p and $\overline{d_j}$ are elected. Finally they are satisfied with respect to any I_t, for $t \neq p, j$, since they are indifferent.

- The $2(m-2)L$ dummy voters whose dependency graph consists of the edge (I_j, I_p) or (I_i, I_r) for some $j \neq p, r$. We argue about the voters with the edge (I_j, I_p) as the other case is also identical. These voters are dissatisfied only with respect to issue I_p, since they are voting for $\{d_j : \overline{d_p}\}$ but d_p is elected. They are satisfied with respect to I_j since they are voting in favor of $\overline{d_j}$ which is elected. Finally they are indifferent with respect to other issues.

- The $2L$ dummy voters whose dependency graph consists of the edge (I_p, I_r) or (I_r, I_p). These voters are dissatisfied with respect to issues I_r and I_p. Consider the ones with the edge (I_p, I_r). They are voting for $\{d_p : \overline{d_r}\}$ but d_r is elected. Additionally, they are voting for $\overline{d_p}$ but d_p is elected. They are satisfied with respect to any other issue since they are indifferent.

The above analysis shows that the dissatisfaction score due to the dummy voters is $(m-2)(m-3)L + 2(m-2)L + 2(m-2)L + 4L = m(m-1)L + 2L$. This is larger by the term $2L$, compared to the dissatisfaction of the dummy voters in outcomes of $\mathtt{POS}_1$. By the choice of L, it is impossible that the dissatisfaction of the non-dummy voters causes an outcome of $\mathtt{POS}_2$ to have a better or equal score than those of $\mathtt{POS}_1$ (note that $L > mn$ and the total dissatisfaction of the non-dummy voters is bounded by mn). Hence, the optimal solution cannot be attained by $\mathtt{POS}_2$. In a similar manner, we can define $\texttt{POS}_k$, for any $k > 2$, and prove that the total dissatisfaction score of any outcome in $\texttt{POS}_1$ is less than the dissatisfaction score of any outcome in $\texttt{POS}_k$. Equivalently, there can be no more than a single positive alternative in the optimal outcome of P''.

Finally, we also need to compare with the "all-negatives" outcome, where $\overline{d_j}$ is selected for every $j \in [m]$. It is a matter of a simple case analysis (like the ones before) to prove that the dissatisfaction incurred by the dummy voters equals $m(m-1)L$, which equals the dissatisfaction of the dummy voters for outcomes in $\texttt{POS}_1$ as well. Hence, now we only need to argue about the non-dummy voters; it is safe to say that these are more than q.

To do this, note that in the original instance P, we can assume without loss of generality that at least one voter approves at least one alternative, otherwise we have a trivial election, where no one expresses any preferences. This means that $x_p > 0$ for at least one issue $p \in [m]$.

By the construction of the preferences for the non-dummy voters in P', the dissatisfaction score for the "all-negatives" outcome equals $\sum_{i \in [m]} x_i$. Recall also that for the outcome in $\texttt{POS}_1$, where only issue p has a positive value, the dissatisfaction of the non-dummy voters is $\sum_{i \in [m] \setminus \{p\}} x_i$. Therefore, there exists an outcome in $\texttt{POS}_1$ which causes strictly less dissatisfaction to the electorate, and this concludes the proof. ⊠

Let P be a $\texttt{YES}$-instance, i.e., say that there is a set S of at most q voters, the deletion of which causes the election of d_ℓ in P, as the unique winner. Note that for every voter in P there is a corresponding voter in P'. Consider the deletion of the set S' from P' that corresponds exactly to the voters of S from P. By exploiting Claim 3.6, it suffices to prove that by deleting S', the outcome $p_\ell = \{\overline{d_1}, \overline{d_2}, \dots, \overline{d_{\ell-1}}, d_\ell, \overline{d_{\ell+1}}, \dots, \overline{d_m}\}$ causes strictly less dissatisfactions in the electorate than $p_j = \{\overline{d_1}, \overline{d_2}, \dots, \overline{d_{j-1}}, d_j, \overline{d_{j+1}}, \dots, \overline{d_m}\}$, for any $j \neq \ell$. To do so, for any $i \in [m]$, let x_i be the number of voters (among the remaining ones, after deleting the set S) who had d_i in their approval list in P. By using the same argument as in the proof of Claim 3.6, we can see that for the non-dummy voters of P' who correspond to those of P, the number of dissatisfactions caused by the outcome p_ℓ in P' is $(\sum_{i \in [m] \setminus \ell, j} x_i) + x_j$.

Also, by doing the same counting argument as in the first part of the proof of Claim 3.6, each dummy voter will be dissatisfied with exactly one issue, and hence they contribute a total of $m(m-1)L$ in the cumulative number of dissatisfactions. In the same manner, we can also have that the total number of dissatisfactions caused by the outcome p_j is $(\sum_{i \in [m] \setminus \{l,j\}} x_i) + x_\ell + m(m-1)L$. Given that d_ℓ was the winning alternative in P after the deletion of voters, it is true that $x_\ell > x_j$ and hence the dissatisfaction caused by p_j is greater than the dissatisfaction caused by d_ℓ. By Claim 3.6, this concludes the forward direction.

For the reverse direction, say that there is a set of voters S' in P', after the deletion of which, d_ℓ is the winning alternative for issue I_ℓ. We note that because of the large value of L, including any dummy voter in the set S' will not influence the final outcome (since $L > q$, even if all the deleted voters are dummy ones, we cannot enforce a different outcome than the outcome without deletions). Hence, we can assume that the set S' has no dummy voters. We can choose then to delete the corresponding set of voters S in P and we will need to prove that, in that case, the elected alternative will be d_ℓ for the single issue of P. By Claim 3.6, and since we assumed that d_ℓ is selected for issue I_ℓ, we know that the optimal solution in the instance after the deletion of S' selected $\overline{d_j}$ for any other $j \neq \ell$. Thus, the winning outcome is $p_\ell = \{\overline{d_1}, \overline{d_2}, \dots, \overline{d_{\ell-1}}, d_\ell, \overline{d_{\ell+1}}, \dots, \overline{d_m}\}$. Therefore, the number of dissatisfactions caused by p_ℓ is lower than the number of dissatisfactions caused by $p_j = \{\overline{d_1}, \overline{d_2}, \dots, \overline{d_{j-1}}, d_j, \overline{d_{j+1}}, \dots, \overline{d_m}\}$, for any $j \neq \ell$. But this implies that $(\sum_{i \in [m] \setminus \{\ell, j\}} x_i) + x_j < (\sum_{i \in [m] \setminus \{\ell, j\}} x_i) + x_\ell$, or equivalently, $x_\ell > x_j$, which means that the number of voters that are not included in S, and have d_ℓ in their approval list, is greater than the number of voters not included in S, who have d_j in their list for any $j \neq \ell$. Thus, in P, d_ℓ will be selected, fulfilling the controller's will.

Adjustments for the CAV-1 reduction: So far we have established that CDV-1 is NP-hard. To prove that CAV-1 is also NP-hard for binary domains, we will perform a similar reduction, this time from CAV-1 with a single issue and with a non-constant domain size (which is again NP-hard by Theorem 3.4). The construction is almost the same, with the difference being that in CAV-1, there are both registered and unregistered voters. Our reduction will assign preferences in the same way as in the proof for CDV-1 and will simply maintain the separation into registered and unregistered voters in the created instance P'. Furthermore, we include the dummy voters in the set of registered voters and we set $L = m(n + q) + 1$. Everything else in the reduction is the same as before, and it is a matter of calculations analogous to the proof of CDV-1 to verify the correctness of the reduction. □

To conclude this subsection, we have now a complete picture for the level of robustness against the malicious actions of adding or deleting voters. Our results act in favor of the CMS rule, showing that a potential controller cannot easily (in terms of computational complexity) enforce her own desirable outcomes, apart from a single case, as shown in Table 3.1. Interestingly, this, single, polynomially solvable case concerns the unconditional setting and when one moves to the conditional case, the considered problem becomes hard for the controller.

Finally, it's worth noting that despite the need for distinct proof approaches in certain cases, the results for the two examined problems (deleting/adding voters) are identical; this observation does not extend to the problems related to the deletion and addition of alternatives, as elucidated in the forthcoming subsection.

3.2 Controlling Alternatives

We now consider the analogous control problems, regarding the addition or deletion of alternatives, instead of voters. It turns out that the picture, from the computational complexity viewpoint, differs sufficiently from the problems considered in the previous subsection.

Instance: A CMS election (I, D, V, B), where $D = D_1 \times \cdots \times D_m$, and D_k is the set of qualified alternatives of each issue I_k, a set D'_k of spoiler alternatives for each I_k (for use only by CAA), an integer quota q, a distinguished alternative $p_j \in D_j$ for a specific issue I_j or an outcome $p \in D$ specifying an alternative for every issue.
Problem CAA-1 (resp. CDA-1): Does there exist a set $D'' \subseteq \cup_{k\in[m]} D'_k$ (resp. $D'' \subset \cup_{k\in[m]} D_k$), with $|D''| \leq q$, such that p_j is the value of the issue I_j in every optimal CMS solution of the profile where the domain of each issue I_k is enlarged by the alternatives in $D'' \cap D'_k$ (resp. reduced by the alternatives in $D'' \cap D'_k$)?
Problem CDA-ALL: Does there exist a set $D'' \subset \cup_{k\in[m]} D_k$, with $|D''| \leq q$, such that p is the unique optimal CMS solution of the profile where the domain of each issue I_k is reduced by the alternatives in $D'' \cap D_k$?
Note: For CDA-1 and CDA-ALL, we also require that for every k, $|D_k \setminus D''| \geq 1$.

Proposition 3.7 *Unconditional Minisum, with arbitrary domain size is immune to CAA-1. For the same setting, CDA-1 and CDA-ALL can be solved in polynomial time.*

Proof. To solve CDA-1 and CDA-ALL we only have to observe that to control a single issue by deleting alternatives in the unconditional case, one can check if the quota is large enough to delete all alternatives that achieve higher approval score than the designated one(s). At what concerns CAA-1, the definition of immunity directly applies, since the controller cannot enforce a designated alternative in Unconditional Minisum by adding some other alternative (whether for the same or for a different issue). □

When we move to instances with conditional ballots, the problems CDA-1 and CDA-ALL do become hard (with the exception of Proposition 3.10 in the sequel). We start with the hardness of CDA-ALL.

Theorem 3.8 *CDA-ALL is NP-hard, when $\Delta_i \leq 1$ for every voter i, and even for a binary domain size in every issue.*

Remark 3.2 We firstly note that all the comments made in Remark 3.1 are applicable here as well. Also, we have not included CAA-ALL in our definitions as CMS is trivially immune to adding spoiler alternatives in order to enforce a qualified alternative in every issue. Concerning the problem CAA-1, we assume that the voters in B may express an opinion about any outcome of every issue, whether it is a qualified one or a spoiler. Additionally, another way to define such problems would be to allow the controller to completely delete or add issues instead of just alternatives. However, given the existence of dependency graphs, erasing an issue can make the preferences of a voter ill-defined. Lastly, the constraint that $|D_k \setminus D''| \geq 1$, for CDA-1 and CDA-ALL, is to ensure that the controller cannot eliminate all the alternatives of an issue.

Proof. Let $P = (G, k)$ be an instance of VERTEX COVER, for an undirected graph G with n vertices and m edges, and an upper bound k on the desirable size of the cover. We denote by $e_1, e_2, \dots, e_m$, the edges of the graph. We will present a reduction from P to an instance P' of CDA-ALL. Let there be the following $2m(k+1)+n$ issues:

- For every edge e_ℓ of G, $\ell \in [m]$, and for $j \in [k+1]$ let $I_{\ell,j}$ and $I'_{\ell,j}$ be a pair of binary issues of domain $\{e_{\ell,j}, \overline{e_{\ell,j}}\}$ and $\{e'_{\ell,j}, \overline{e'_{\ell,j}}\}$ respectively,. We refer to these issues as edge issues.
- For every vertex v of G let I_v be a binary issue of domain $\{d_v, \overline{d_v}\}$. We refer to these issues as vertex issues.

Note that, we have added one vertex issue for every vertex of G and, essentially, we have introduced two edge issues for every edge of G, but in $(k+1)$ "copies". This will play a significant role in the reverse direction of the reduction. The voters of the created instance P' will be $2m+3$ in total, and we partition them as follows:

- Group 1: There are 2 edge voters for every edge e_i of G. For an edge $e_i = (u, v)$ and for every $j \in [k+1]$, these voters submit the same ballot $\{d_u : e_{i,j}, d_v : e'_{i,j}\}$ (where we have arbitrarily chosen that the satisfaction of one edge issue depends on the vertex issue I_u and the other edge issue depends on the remaining vertex issue I_v). The voters are indifferent for the rest of the issues. Hence, every such voter is interested in exactly $2(k+1)$ edge issues, each conditioned on either I_u or I_v. At what follows, we will refer to these voters as "group 1" or edge voters.
- Group 2: There are 2 voters who are voting for $\{\overline{e_{i,j}} : \overline{e'_{i,j}}\}$, for every pair $(i, j) \in [m] \times [k+1]$, and who are indifferent for every other issue. At what follows, we will refer to these voters as "group 2".
- There is one voter, that we will refer to as the special voter, who is voting unconditionally $\overline{d_1}, \overline{d_2}, \dots, \overline{d_n}$ for the vertex issues, and is indifferent for every other issue.

To complete the construction of the instance P', we use k as the quota parameter, and we suppose the controller wants to enforce the outcome

$$(\overline{e_{i,j}}, \overline{e'_{i,j}})_{i \in [m], j \in [k+1]}, (\overline{d_j})_{j \in [n]},$$

by removing at most k alternatives. It is trivial to observe that all the issues in the instance P' are of binary domain and that for every voter i, $\Delta_i \leq 1$ in her dependency graph.

Before we proceed, we note that the designated outcome fully satisfies the voters from group 2 as well as the special voter, however, it dissatisfies every edge voter with respect to all $2(k+1)$ issues they care about each, hence it produces a total dissatisfaction score of $4m(k+1)$. Additionally, we highlight that there are also other outcomes with the same score, which prevent the designated one from being the unique winner without deleting any alternatives in P'. For instance, if e_t is the edge (u, v) of G, then consider the following outcome, for any $\ell \in [k+1]$:

$$\{(e_{t,\ell}, e'_{t,\ell}), (\overline{e_{i,j}}, \overline{e'_{i,j}})_{(i,j)\in[m]\times[k+1]\setminus(t,\ell)}, du, dv, (\overline{d_j})_{j\in[n]\setminus\{u,v\}}\} \tag{3.1}$$

The outcome described in Equation (3.1) has a dissatisfaction score of $4(m-1)(k+1)+4k$ from group 1 and a dissatisfaction score of 1 from each voter in group 2 and a dissatisfaction score of 2 for each remaining voter, leading to a total score of $4m(k+1)$.

To see now the forward direction of the reduction, suppose there exists a vertex cover in G of size at most k, which is formed say by a set of vertices $S \subset [n]$. Then we choose to delete in the created instance P' of CDA-ALL, the corresponding positive alternatives $\{d_j\}$ for $j \in S$, from the vertex issues $\{I_j\}_{j\in S}$, respectively. Hence, any solution to the resulting instance after these deletions, should now definitely contain the alternative $\overline{d_j}$ for every issue I_j, such that $j \in S$. Clearly, the designated outcome still remains a valid solution with a total score of $4m(k+1)$. We need to see what happens with the rest of the possible outcomes.

One can easily verify that the best solution among the feasible ones in which all edge voters are dissatisfied with respect to all the issues they care about, is precisely the designated outcome (because it satisfies in all issues all the other voters from group 2 as well as the special voter, and also no other solution can achieve the same). Hence, it remains to see if there exists any optimal solution with at least one pair of edge voters satisfied with respect to at least one issue.

For the sake of contradiction, assume that this is the case, and consider a pair of such edge voters, corresponding to edge $e = (u, v)$. Without loss of generality, suppose that u does not belong to the vertex cover, and hence d_u is an available alternative, that has been selected in the optimal solution we are considering, and also that d_v has been deleted. This is due to the vertex cover property, implying that either d_u or d_v has been deleted and furthermore, if both had been deleted, the pair of edge voters that we focus on, would not be satisfied with respect to any issue, contrary to what we have assumed. The selection of d_u may also cause other edge voters (whose edge is incident with u) to be satisfied as well, with respect to some issues.

To arrive at a contradiction, and for ease of notation, let $e_1, \dots, e_\ell$, be the edges that have u as an endpoint, and that correspond to edge voters who are satisfied with respect to at least one of the issues they care about, under the optimal solution we are considering. Additionally, for such an edge e_i, let $m(i)$ be the number of issues that correspond to e_i, and with respect to which, the two edge voters of e_i are satisfied (this occurs because the edge voters may have declared $d_u : e_{i,j}$ or $d_u : e'_{i,j}$). Clearly $m(i) \leq k+1$. Counting the total number of satisfactions of edge voters due to the selection of d_u, we get exactly $2\sum_{i=1}^{\ell} m(i)$. But if we now replace d_u by $\overline{d_u}$ and set all the edge issues of the edges $e_1, \dots, e_\ell$, to their negative value, then we will dissatisfy all the edge voters that we were satisfying before, but we get $2\sum_{i=1}^{\ell} m(i)$ new satisfactions from group 2 and one new satisfaction from the special voter. Therefore, we reach an outcome with a lower dissatisfaction score, contradicting the fact that we started with an optimal solution.

We conclude that there cannot exist an optimal solution, after the deletions we made, where some of the voters from group 1 enjoy any satisfaction. Therefore, we can only satisfy group 2 and the special voter, and we can conclude that after the deletion of at most k (positive) alternatives that correspond to a vertex cover, the designated outcome becomes the unique winner.

For the reverse direction, suppose that there is a set D of at most k alternatives, the deletion of which, forces the designated outcome to be the unique optimal solution. Trivially, D cannot contain negative values neither from edge nor from vertex issues. We denote by D_V the subset of D that contains positive values from vertex issues and let S be the corresponding set of vertices in G. We claim that S forms a vertex cover of G. Towards a contradiction, say that S is not a vertex cover. Then, there exists an edge $e_t = (u, v)$ such that both $d_u, d_v \notin D_V$, thus d_u and d_v are still feasible alternatives, after the deletions we have made. Note that due to the budget constraint, it holds that $|D''| \leq k$. Hence out of the $2(k+1)$ issues that the edge voters corresponding to e_t care about, there exists an index ℓ, such that both $e_{t,\ell}$ and $e'_{t,\ell}$ are still available, after the deletion of D. But then, the outcome described in Equation (3.1) is still feasible, which contradicts the fact that the designated outcome became the unique winner after the deletion of D. □

Moving to CDA-1 and CAA-1 we show that we can have hardness results, but only under a non-constant domain size for at least one issue. The proof of Theorem 3.9 below, shows a connection with some natural problems on graphs, that have been previously linked to election control for other voting rules [BU09].

[BU09] Betzler and Uhlmann (2009): Parameterized Complexity of Candidate Control in Elections and Related Digraph Problems.

Theorem 3.9 *CAA-1 and CDA-1 are NP-hard, when $\Delta_i \leq 1$ for every voter i, and even when the treewidth of the global dependency graph is at most one, but with non-constant domain size in at least one issue.*

Proof. We will firstly prove the hardness of CDA-1. We will perform a reduction from the NP-hard problem MAX OUT-DEGREE DELETION (MOD) [BU09].

In MAX OUT-DEGREE DELETION (MOD) we are given a directed graph $G = (V, E)$, a special vertex $p \in V$ and an integer $k \geq 1$ and the goal is to determine whether there exists a set $V' \subseteq V$ with $|V'| \leq k$ such that p is the only vertex of maximum out-degree in $G[V \setminus V']$.

For $S \subseteq V$, we denote by $deg_S(u)$ the out-degree of vertex u in a graph $G = (V, E)$, when we count only outgoing edges towards the vertices of S. Let $P = (G = (V, E), p, k)$ be an instance of MOD in a directed graph with n vertices and m edges We create a CDA-1 instance, where we have one issue I_j for every vertex $v_j, j \in [n]$ and an extra issue I_0, hence $I = \{I_0, I_1, I_2, \dots, I_n\}$. For $j \in [n]$, the domain of issue I_j is binary in the form $D_j = \{d_j, \overline{d_j}\}$. The domain of I_0, say D_0, contains $(k+1)(n-1)+1$ alternatives. In particular, it contains an alternative b_p that corresponds to the designated vertex $p \in V$, and for every vertex $v \in V \setminus \{p\}$, there are $k+1$ alternatives b_v^ℓ, for $\ell \in [k+1]$. Essentially, these are identical $k+1$ 'copies' encoding the selection of v in I_0, and play a significant role in the reverse direction of the reduction. As for the voters, there are two types of voters, *edge voters* and *vertex voters*. There is one edge voter for every edge $(i, j) \in E$, with a dependency graph having one edge from I_j to I_0, and voting as follows:

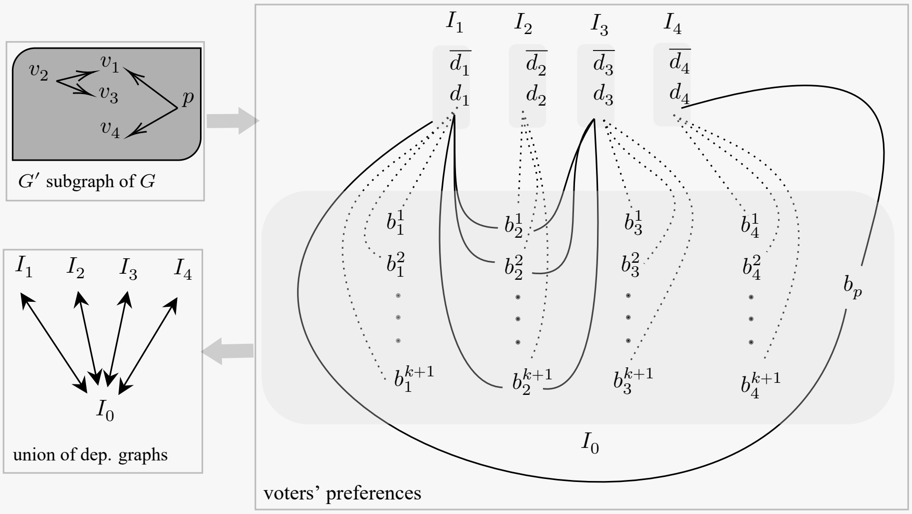
- ➛ For the issue I_0, she votes conditioned on I_j for $\{d_j : b_i\}$ if $i = p$ or otherwise for $\{d_j : b_i^\ell\}, \forall \ell \in [k+1]$.
- ➛ For all other issues she is satisfied with any alternative.

For every vertex other than p, we also have a block of L identical voters, where it suffices to take $L = m + 1$. Each voter in the j-th block, with $j \in V \setminus \{p\}$ has a dependency graph with 1 edge, from I_0 to I_j and votes as follows:

- ➛ For the issue I_j, she is satisfied with the combinations $\{b_j^\ell : d_j\}$ for any ℓ. Also, if the value of I_0 differs from b_j^ℓ, for any ℓ, she is satisfied with any value on I_j. Hence, the only restriction is that when the value of I_0 comes from an alternative corresponding to vertex j, the voter can be satisfied with respect to I_j only by d_j.
- ➛ For all other issues, she is satisfied with any alternative.

In total, we have $m + (n-1)L$ voters. We also use k as the quota parameter, and we suppose the controller wants to enforce the alternative b_p at issue I_0. Clearly, for every voter i, $\Delta_i \leq 1$ in her dependency graph, and the global dependency graph is a star centered on I_0. The maximum domain cardinality is $O(kn) = O(n^2)$.

For a better view of the construction we comment on the figure below, which illustrates the reduction to CDA-1.


v_2
v_1
v_3
p
v_4
G' subgraph of G
I_1 I_2 I_3 I_4
I_0
union of dep. graphs
I_1 I_2 I_3 I_4
$\overline{d_1}$ $\overline{d_2}$ $\overline{d_3}$ $\overline{d_4}$
d_1 d_2 d_3 d_4
b_1^1 b_2^1 b_3^1 b_4^1
b_1^2 b_2^2 b_3^2 b_4^2
b_p
b_1^{k+1} b_2^{k+1} b_3^{k+1} b_4^{k+1}
I_0
voters' preferences

In particular, it illustrates only a part of the construction that pertains to the vertices of a subgraph G' of the initial graph G given in the instance of MOD (as shown in the upper-left part of the figure). The figure also depicts the voters' ballots (rightmost part of the figure) and the global dependency graph which emerges (lower-left part of the figure). To be more precise, the lower-left part shows the union of the dependency graphs of all voters, where both orientations are present for the edges shown. Hence, the global dependency graph is simply a star centered on I_0. The connections in the rightmost part of the figure represent acceptable pairs of alternatives by voters. More precisely, a dotted connection between the alternatives d_j and b_j^ℓ for some j and ℓ, represents the conditional approval ballot $\{b_j^\ell : d_j\}$ of the block of the L identical vertex voters that correspond to v_j of G'. A solid connection between the alternatives d_j and b_i^ℓ (resp. between d_i and b_p) represents the conditional approval ballot $\{d_j : b_i^\ell\}$ (resp. $\{d_j : b_p\}$) of an edge voter corresponding to edge (v_i, v_j) (resp. (p, v_j)) of G'.

Suppose there exists a set S of vertices in G of size at most k, say without loss of generality that $S = \{1, \ldots, k\} \subseteq V$, whose deletion leaves p as the only vertex of maximum out-degree. We now choose to delete the corresponding alternatives $\{d_1, \ldots, d_k\}$ from the issues $\{I_1, \ldots, I_k\}$. If we select b_p for the issue I_0, then the total dissatisfaction score can be brought down to $m - deg_{V \setminus S}(p)$ by choosing d_j for every issue I_j where d_j has not been deleted. To see this, the only edge voters that are satisfied with respect to I_0 are edges that are outgoing from p and whose other endpoint belongs to $V \setminus S$. Hence all remaining $m - deg_{V \setminus S}(p)$ voters will be dissatisfied with respect to I_0. Regarding the vertex voters, they will all be satisfied on all issues.

On the other hand, if we select for I_0 some b_j^ℓ for any $\ell \in [k+1]$, we need to consider two cases, depending on j. If $j \in V \setminus S$, then by the same reasoning as before, the best we could achieve is to have a dissatisfaction score equal to $m - deg_{V \setminus S}(j)$. But since p has the maximum out-degree, this would yield a worse solution. Now suppose $j \in S$. Then we know that d_j has been deleted from I_j. Hence, the j-th block of vertex voters will be dissatisfied with respect to I_j, and since $L > m$, this cannot yield an optimal solution. To conclude, after the deletion of the selected alternatives, b_p has to be selected for I_0 in any optimal solution.

For the reverse direction, suppose that there is a set D'' of at most k alternatives, the deletion of which, forces b_p to be selected for I_0 in every optimal solution. It is without loss of generality to assume that D'' does not contain anything from D_0. To elaborate on this claim, since there are $k + 1$ copies of alternatives for every $i \in V \setminus \{p\}$ that have an identical role, there is no change in the optimal outcome by deleting up to k alternatives from I_0 (some alternative will survive for every i). Moreover, we can assume that none of the deleted alternatives equals $\overline{d_j}$ for some issue $I_j \neq I_0$ since if it were, we can swap it with d_j without harming the cost of the optimal solution (one cannot strengthen the support of b_p in I_0 by deleting $\overline{d_j}$ for some j). Also, bear in mind that we are not allowed to delete both d_j and $\overline{d_j}$ from an issue $I_j, j \in [n]$, as there are no other choices left for I_j.

To summarize, the deleted alternatives must come from distinct issues among $I_1, \dots, I_n$ and they all correspond to some d_j for $j \in [n]$. It is now easy to observe that deleting from V the set S formed by the vertices corresponding to these alternatives in D'', makes p the unique vertex of maximum out-degree in the induced subgraph of G. If not, there is a vertex, say $v \in V \setminus S$, with greater or equal out-degree. In that case, if we select $b_v^{\ell'}$ for I_0 for some arbitrary ℓ', and d_j for all issues I_j, for which d_j has not been deleted, we will obtain a solution with at most the same dissatisfaction score as the optimal solution that used b_p. Indeed, we will have fewer or equal dissatisfactions from the edge voters with respect to I_0, and also all the blocks of the vertex voters will be satisfied (the block of voters who care about I_v is satisfied because d_v has not been deleted, since $v \in V \setminus S$). This contradicts the fact that b_p was elected for I_0 in every optimal solution.

For the NP-hardness of CAA-1, the proof is based on a similar reasoning as in the proof of CDA-1, but with appropriate adjustments. First, it is more convenient to perform a reduction from a slightly different problem, which is the MAX-OUTDEGREE ADDITION (MOA) problem.

In MAX-OUTDEGREE ADDITION (MOA) [BU09] we are given a directed graph $G = (V_1 \cup V_2, E)$, where V_1 denotes the set of registered vertices, and V_2 is the set of unregistered vertices, a distinguished vertex $p \in V_1$ and an integer $k \geq 1$ and the goal is to determine whether there exists a set $V' \subseteq V_2$ with $|V'| \leq k$ such that p is the only vertex that has maximum outdegree in $G[V_1 \cup V']$.

Starting from an instance of MOA, where $n = |V_1| + |V_2|$, let $I = \{I_0, I_1, I_2, \dots, I_n\}$. For $j \in V_1$, we have two qualified alternatives, $D_j = \{d_j, \overline{d_j}\}$ and no spoiler ones. For $j \in V_2$, we have one qualified alternative,[1] $D_j = \{\overline{d_j}\}$, and we will have d_j as a spoiler alternative, $D'_j = \{d_j\}$. The domain of I_0 corresponds to all the vertices and equals $D_0 = \{b_1, \dots, b_n\}$. In contrast to CDV-1, we do not need to have $k + 1$ "copies" for each b_i, since the spoiler alternatives that will be added are not going to be from D_0. As for the voters,

[1] If one wishes to avoid issues with unary starting domain, we can also add one dummy qualified alternative, so that no issue is trivialized before the addition of any spoiler alternatives.

there is one edge voter for every edge of the graph, regardless of whether its endpoints belong to V_1 or V_2 and one vertex voter for every vertex of the graph. All voters have similar preferences as in the CDA-1 reduction, from which their ballots for each issue I_j with $j \in \{0, 1, \dots, n\}$ can be immediately obtained by replacing, $\{b_j^\ell\}_{\forall \ell \in [k+1]}$ with b_j. For example, an edge voter arising from an edge (i, j) will vote for the combination $\{d_j : b_i\}$ regarding I_0. Using similar arguments as in the proof for CDA-1, we conclude that there is a way to add up to k vertices and make p the unique vertex with maximum out-degree if and only if there is a set of at most k alternatives to add in the CAA-1 instance to fulfill controller's will. □

Notably, moving to a constant domain size, the considered problems, CDA-1 and CAA-1, seem to behave differently, as the following result indicates.

Proposition 3.10 *CDA-1 can be solved in polynomial time, when $\Delta_i \leq 1$ for every voter i, the treewidth of the global dependency graph is constant and the domain size is also constant for every issue.*

Proof. Let q be the quota parameter and let I_j be the issue where the controller wants to enforce a specific alternative. If $q \geq |D_j| - 1$, then we can simply delete precisely all other $|D_j| - 1$ alternatives of I_j so that the controller's will is the only choice left. If $q < |D_j| - 1$, this implies that $q = O(1)$. Then we can check all possible ways of picking up to q items from the available set of alternatives of all issues (polynomial in m). For every such combination, since the conditions of Theorem 2.1 hold, we solve the remaining instance checking if the controller's choice appears in all optimal solutions, by solving CMS with and without the designated alternative. □

Hence, a constant domain size makes a difference for CDA-1 when we stick to the assumptions from Section 2.2 on each Δ_i and on the treewidth. For CAA-1, we are not yet aware if the same result holds (the proof arguments certainly do not go through), and we leave this as an interesting open problem. However, we have established intractability, as soon as we move to slightly richer instances with $\Delta_i \leq 2$.

Theorem 3.11 *CAA-1 is NP-hard, when $\Delta_i \leq 2$ for every voter i, even when the treewidth of the global dependency graph is at most one and even for a binary domain size in every issue.*

Proof. Consider an instance P of VERTEX COVER, asking if there is a cover of size at most k in a graph $G = (V, E)$, with $|V| = n$, $|E| = m$. We create an instance P' of CAA-1 with $n + 1$ issues $I = \{I_0, I_1, \dots, I_n\}$. The issue I_0 has two qualified alternatives, $D_0 = \{d_0, \overline{d_0}\}$. Each issue I_j for $j \in [n]$ corresponds to a vertex od G, and has one qualified alternative, denoted by $\overline{d_j}$, and one unqualified one denoted by d_j. Formally, $D_j = \{\overline{d_j}\}$ and $D'_j = \{d_j\}$, for $j \in [n]$. As for the voters, we have a total of $2m - 1$ voters. The first m voters correspond to the edges of G, and they are satisfied with all the alternatives in the issues I_j, $j \in [n]$. For issue I_0, each edge voter has a dependence on the two issues corresponding to its endpoints. In particular, for an edge (j, ℓ), the corresponding edge voter has a dependence of I_0 on both I_j and I_ℓ. He is satisfied with respect to I_0, only when either d_j or d_ℓ is selected, and d_0 is selected as well. Thus he is satisfied with the combinations $\{(d_j, x) : d_0\}$ for any $x \in \{d_\ell, \overline{d_\ell}\}$, and with $\{(x, d_\ell) : d_0\}$ for any $x \in \{d_j, \overline{d_j}\}$. These together encode precisely the constraint $(d_j \vee d_\ell) : d_0$. Any other combination of alternatives of I_j, I_ℓ, and I_0 make this edge voter dissatisfied with respect to I_0. The remaining $m-1$ dummy voters are satisfied with all the alternatives of the first n issues and are also satisfied only with $\overline{d_0}$ for issue I_0. To complete the construction, we use k from P as the quota of P', and we assume that the controller wants to enforce d_0 on issue I_0. It is easy to check that the maximum in-degree for every voter is at most two, and that the global dependency graph is a star centered on I_0, and hence with treewidth equal to one.

Suppose that P has a vertex cover S of size at most k. We then add in P' the unqualified alternatives for the issues that belong to the vertex cover of G. By selecting those alternatives, and with d_0 for I_0, and any alternative for the remaining issues, we claim that all the edge voters are satisfied with respect to I_0 (since for every edge, at least one of the added alternatives together with d_0 satisfy the corresponding constraint). Thus, there is only 1 unit of dissatisfaction from every dummy voter on I_0, with a total score of $m - 1$. Any solution where d_0 is not the selected choice for I_0 would dissatisfy all the edge voters, and would have a score of at least m, hence cannot be optimal. Thus, in every optimal solution, I_0 is assigned the value of d_0.

For the reverse direction, suppose that there is a set of at most k unqualified alternatives that, when added, ensure that d_0 is selected in every optimal solution. We know that selecting d_0 causes the dummy voters to be dissatisfied, hence the optimal dissatisfaction score is at least $m-1$. If $\overline{d_0}$ was chosen for I_0, we know that the total dissatisfaction score is m (due to the edge voters), and since this cannot be optimal, we have that the dissatisfaction score in an optimal solution is exactly $m-1$. But this means that all the remaining m voters, or equivalently all edge voters, have to be satisfied with all issues in the optimal solution, i.e., satisfied with I_0 as well. Thus, the added alternatives need to satisfy every edge voter, so if a voter's dependence of I_0 is based on issues I_j and I_ℓ, then either d_j or d_ℓ has been added (or both), and hence the set of added alternatives correspond to a vertex cover of size at most k. □

Overall, we have shown that CMS is indeed sufficiently robust against malicious actions, in most of the variants of the control problem considered. The behavior regarding the addition or deletion of alternatives does exhibit differences with that of deleting or adding voters. In some cases, such as in adding alternatives, we get the stronger guarantee of immunity, compared to NP-hardness. On the other hand, when deleting alternatives, we have vulnerability in more settings than when deleting voters. Still, we do have computational hardness, when the domain size is large enough.

3.3 Concluding Discussion and Future Directions

In the current chapter we focused on computational aspects of CMS elections from the perspective of strategic attempts to control election outcomes. We have concluded that CMS exhibits sufficient robustness against control actions in the considered settings. There are still several interesting problems for future research: There is one case of our control questions that has been left open, namely, the complexity of CAA-1, under a binary domain size and with $\Delta_i \leq 1$. Even more interestingly, one can study other strategic moves such as destructive versions of control or bribery in a CMS election, or consider a study on instances that are not worst-case. Along the same spirit, CMS was proven to be non-strategyproof by [BL16], but the complexity of manipulating has not been examined yet.

Guarantees for Proportional Representation in Conditional Approval Voting

4

Proportional representation of voters' preferences is an important desideratum in a wide range of Social Choice settings, including parliamentary elections [Puk14], approval-based multiwinner voting [LS23], and participatory budgeting [PPS21]. In two recent papers, Freeman et al. [FKP21] and Skowron and Górecki [SG22] studied a *public decision* model, where a collective decision needs to be taken on a given set of unrelated binary (yes/no) issues. In real life scenarios, these kinds of situations are often handled via majority rule: an issue is implemented if and only if more than 50% of the voters are in favor of it; and while this might be the most straightforward decision rule, it comes at a disadvantage: It is not proportional. For instance, consider a scenario where three binary decisions need to be taken, and the electorate is split into two groups. The first group of voters, which makes up two thirds of the electorate, votes "yes" on every issue, while the second group votes "no" on all issues. Majority rule would set all issues to "yes", completely ignoring the preferences of one third of the electorate. One might argue that setting two issues to "yes" and one issue to "no" would be a more fair solution. Such considerations led Freeman et al. [FKP21] and Skowron and Górecki [SG22] to adopt several proportionality notions and proportional voting rules from the setting of approval-based multiwinner voting to their model.

[FKP21] Freeman et al. (2021): Proportionality in Approval-Based Elections with a Variable Number of Winners.

[SG22] Skowron and Górecki (2022): Proportional Public Decisions.

The model studied by the above-mentioned previous works has two important limitations. Firstly, the issues under consideration are only allowed to be binary, which makes it impossible to model scenarios where an issue has more than two possible alternatives. Secondly, their setting does not allow any dependencies between the issues, although in some situations it might be the case that the approval of an issue is dependent on the decision taken for another. The more general setting for proportional decision-making that we study here is based on the *conditional approval voting* framework introduced by Barrot and Lang [BL16]. As highlighted in Chapter 2, this setting allows for more than two choices per issue as well as for dependencies between the issues; consequently, it addresses both limitations discussed above. The following example illustrates the type of situations that the model can deal with, within the context of public decisions.

[BL16] Barrot and Lang (2016): Conditional and Sequential Approval Voting on Combinatorial Domains.

Example 4.1 Consider a scenario, where a municipality has decided that the following projects will be funded: a public park, a pedestrian infrastructure, and a community center. However, the location of these sites is not fixed yet, and the voters should decide on whether to build each project at the Southside, the Centralside, or the Northside district. Obviously, the domain of each issue here is of size three. Voter 1, a resident of Southside, is voting in favor of building the park in her neighborhood and therefore casts an (unconditional) approval ballot for this option. The preferences of the other voters are more complicated: Voter 2 is concerned about the traffic congestion that a new community center could cause and votes for building a community center at any location only if a pedestrian infrastructure is built at the same location as well. Finally, voter 3 doesn't want any two projects built in the same location. Hence her approval for any project depends on the decisions made for the other two projects.

In this chapter, we focus on the proportionality concept that was introduced by Skowron and Górecki [SG22], which provides strong guarantees for all groups of voters of sufficient size. These guarantees are considered as more powerful than other previously studied ones, which were only able to ensure proportional representation to groups of voters that have similar preferences. Such "cohesive" groups are rare to be found in practice in the unconditional setting [BFN+19], let alone when dependencies between issues exist.

[BFN+19] Bredereck et al. (2019): An Experimental View on Committees Providing Justified Representation.

Contribution.

The main challenge left open in [SG22] lies in incorporating in their model dependencies between issues, which can be seen as a stepping-stone to the building of a theory of fairness for general public decision settings. Motivated by this, we first introduce appropriate generalizations of *Proportional Approval Voting* (PAV) and the *Method of Equal Shares* (MES), which are two prominent rules with provable guarantees on proportional representation for binary, unrelated issues. We then make progress along two fronts. First, we generalize the known guarantees to elections over issues with non-binary domains. Secondly, and most importantly, we consider elections with dependencies among issues, where we identify sufficient restrictions that lead to analogous results. Our work is the first that provides guarantees of proportionality for elections with conditional ballots. Moreover, for the conditional version of MES, we also study computational aspects. Our results demonstrate that the conditional setting poses additional challenges and differs significantly from the unconditional one, both in terms of proportionality and complexity. As an example, we show that MES is hard to implement in the general conditional case, whereas it has been known to be tractable in unconditional elections. Another highlight of the current chapter is that PAV and MES achieve proportionality bounds under different assumptions (that have some degree of complementarity), and thus, one cannot reach yet an absolute conclusion when compared against each other, under the conditional framework, in contrast to the unconditional setting.

Related Work.

On a high level, there are three lines of research that are related to the work presented in current chapter. First, our work is closely related to approval-based multiwinner voting [LS23]. Starting with the works of Aziz et al. [ABC+17] and Sánchez-Fernández et al. [SEL+17], a plethora of proportionality notions have been defined and studied. Originally defined for multiwinner voting, proportionality notions have more recently been extended to more general settings such as participatory budgeting [BFL+23; PPS21]. Quantitative proportionality notions have also been studied by Aziz et al. [AEH+18] and Skowron [Sko21].

Secondly, our work is related to multi-issue decision-making and seat-posted elections. With the exception of the aforementioned papers by Skowron and Górecki [SG22] and Freeman et al. [FKP21], these have not been studied from a proportionality perspective. Instead, the literature has focused on issues such as Anscombe's and Ostrogorski's paradoxes [Ans76; FW22a; LL06] or axiomatic comparisons between preferences and decisions over issues and the entire decision space [ADL22; DL22].

[SG22] Skowron and Górecki (2022): Proportional Public Decisions.

[FKP21] Freeman et al. (2021): Proportionality in Approval-Based Elections with a Variable Number of Winners.

Third, our work is utilizing a model of conditional approval voting over combinatorial domains. This was introduced by Barrot and Lang [BL16], who also proposed three voting rules for incorporating dependencies between issues and studied ()mainly) axiomatic properties. Later on, the works by Markakis and Papasotiropoulos [MP20; MP21b], focused on the algorithmic aspects of the winner determination problem under the conditional analog of the (Minisum) approval voting rule (see Chapter 2) and also considered the computational complexity of controlling the election outcome, under the same rule (see Chapter 3). Apart from approval-based elections, the presence of preferential dependencies remains a major challenge and several frameworks have been considered, as extensively discussed by Lang and Xia [LX16] and Chevaleyre et al. [CEL+08].

[BL16] Barrot and Lang (2016): Conditional and Sequential Approval Voting on Combinatorial Domains.

[MP20] Markakis and Papasotiropoulos (2020): Computational Aspects of Conditional Minisum Approval Voting in Elections with Interdependent Issues.

[MP21b] Markakis and Papasotiropoulos (2021): Winner Determination and Strategic Control in Conditional Approval Voting.

[LX16] Lang and Xia (2016): Voting in Combinatorial Domains.

[CEL+08] Chevaleyre et al. (2008): Preference Handling in Combinatorial Domains: From AI to Social Choice.

4.1 Preliminaries

In this section, following a brief discussion on the setting of conditional approval voting (extensively presented in Section 2.1), we define the proportionality notion we are interested in, and define the two voting rules that will be the focus of the current chapter.

4.1.1 Conditional Election Setting

In describing conditional approval elections, we closely adhere to the notation and terminology presented in Section 2.1, with some minor adjustments. The introduction of a few additional definitions is being discussed below.

Recall that a *conditional approval voting instance*, or simply an *instance*, is determined by a tuple $P = (I, N, B)$, where B denotes the conditional approval ballots of the set of voters N over the issues of I, as described in details in Section 2.1. Here we define the group of voters as $N = \{v_1, v_2, \dots, v_n\}$ and we say that each issue I_j, is associated with a *domain* $D_j = \{a_j^1, a_j^2, \dots, a_j^d\}$ that corresponds to the alternatives for this issue. Without loss of generality, we assume that $|D_j| = |D_{j'}| \geq 2, \forall j, j' \in [m]$ (if not, one can add dummy alternatives) and we denote the domain size by d. We use $\Gamma_i(I_j)$ to denote the (possibly empty) set of direct predecessors, i.e., in-neighbors, of issue I_j in G_i and $\Gamma_i^*(I_j) = \Gamma_i(I_j) \cup \{I_j\}$. Therefore, voters cast conditional approval ballots that are expressed as follows: For an issue I_j with $|\Gamma_i(I_j)| = 0$, voter v_i casts a standard (unconditional) approval ballot, stating explicitly all the alternatives of D_j that are approved by her, the number of which varies from 0 to d. For the case that $|\Gamma_i(I_j)| > 0$, voter v_i needs to specify all the combinations of alternatives for issues in $\Gamma_i^*(I_j)$ that she approves, i.e., that make her satisfied with respect to issue I_j. These combinations are expressed in the form $\{s : t\}$, where $s \in \times_{I_\ell \in \Gamma_i(I_j)} D_\ell$, and $t \subseteq D_j$. Such a ballot signifies the satisfaction (i.e., approval) of a voter with respect to issue I_j, when the in-neighbors of I_j in G_i are set to the alternatives specified by s, and the selection for I_j belongs to t. We note that we do not impose that voters submit a ballot for every issue (they can abstain if they are not satisfied with any outcome with respect to certain issues). In the case that all voters have the same dependency graph, we let $\Delta_{\texttt{in}}$ denote the maximum in-degree of this graph.

We use $\Gamma(I_j)$ to denote the (possibly empty) set of neighbors of issue I_j in G and $\Gamma^*(I_j) = \Gamma(I_j) \cup \{I_j\}$. Moreover, recall that Δ denotes the maximum degree of any vertex in G. Note that if $\Gamma(I_j) = \emptyset$ for all $j \in [m]$, then a conditional approval voting election degenerates to a classic approval election over m issues.

For any $I' \subseteq I$, any tuple $w \in \times_{I_j \in I'} D_j$, i.e., that includes an alternative from D_j for every issue $I_j \in I'$, is referred to as a *suboutcome*. If w specifies a value for all issues of an instance $P = (I, N, B)$, we simply call it an *outcome* of P. A *conditional approval voting rule* is a function that maps each conditional approval voting instance P to an outcome w. Given an outcome w, we let $u_i(w)$ denote the number of issues with respect to which voter v_i is satisfied under w. If w is clear from the context, we simply refer to this as u_i.

4.1.2 Proportionality Criterion

We now define a (parameterized) notion of proportionality that generalizes the one suggested by Skowron and Górecki [SG22]. The basic rationale behind any proportionality definition is the intuitive idea that any fraction of voters should have the ability to influence a corresponding fraction of decisions:

[SG22] Skowron and Górecki (2022): Proportional Public Decisions.

> *A group of voters that makes up a β-fraction of the electorate should be able to decide on $\beta \cdot m$ issues.*

In particular, such a criterion requires that minority opinions are represented as well (proportionally to their size). To formalize this notion, an important parameter is the set of issues that a voter can be possibly satisfied with. Namely, for a voter $v_i \in N$, we let R_i denote the set of issues $I_j \in I$, for which v_i has submitted at least one conditional approval ballot (or unconditional if $\Gamma_i(I_j) =$), i.e., there is at least one selection of alternatives for $\Gamma_i^*(I_j)$ that makes v_i satisfied with respect to issue I_j. Formally,

$$R_i = \{I_j \in I : \exists w \in \times_{I_\ell \in \Gamma_i^*(I_j)} D_\ell \text{ that satisfies } v_i \text{ with respect to } I_j\}.$$

By definition, voter v_i cannot be satisfied with respect to any issue in $I \setminus R_i$, under any outcome. For a group $V \subseteq N$ of voters, we let r_V denote the number of issues for which all voters of V approve at least one alternative, i.e., $r_V = |\bigcap_{i \in V} R_i|$. The role of r_V is important in the definition below in which, the proportionality guarantee of a group of voters takes into account the maximal number of issues that all members of the group care about.

To avoid trivialities, we assume that $|R_i| > 0$ for all $v_i \in N$.

Definition 4.1 *A conditional voting rule is α-proportional, for some $\alpha \in [0, 1]$, if for every conditional approval voting instance $P = (I, N, B)$ with $|N| = n$ and for every $V \subseteq N$, there exists a voter $v_i \in V$ such that if w is the winning outcome under the considered rule, then*

$$u_i(w) > \alpha r_V \frac{|V|}{n} - 1.$$

The parameter α in the definition represents the degree of proportionality that a voting rule can guarantee. Ideally, we would like to have α-proportional rules for $\alpha = 1$, as this would mean that the elected outcome aligns with the views of the electorate in a proportional manner. However, as we will soon show, such a rule does not exist and more relaxed values will need to be considered. In the unconditional case, and with a binary domain for each issue, it was shown in [SG22] that $\alpha = \frac{1}{2}$ is achievable. In our more general setting, we will see that the degree of proportionality cannot be expressed by a constant; rather, it will be a function of the input instance, dependent on d and Δ.

4.1.3 Conditional Voting Rules

We focus on two conditional voting rules that constitute natural generalizations of their well-studied unconditional versions. We begin by presenting their unconditional analogs below for ease of reference.

PAV and MES in the Unconditional Setting.

Under *Proportional Approval Voting* (PAV), an outcome w in an unconditional instance $P = (I, N, B)$ gains a score of $\sum_{k=1}^{u_i(w)} \frac{1}{k}$ from every voter v_i of N who is satisfied with w with respect to $u_i(w)$ issues. The outcome that achieves the maximum score is the winning one.

Method of Equal Shares (MES) consists of two phases, and the first one works in rounds. Initially, each voter is given a budget m, equal to the number of issues and all issues are being considered as unfixed. Fixing an issue will cost a price of n. Given that B only contains unconditional ballots, we only allow a single issue to be fixed at any round. In any round t of the first phase we are considering every unfixed issue $I_j \in I$ and for every possible alternative of issue I_j, namely $w \in D_j$, we perform the following: First, we identify the set of voters $S(w)$ who have a positive remaining budget and are satisfied with respect to I_j under w; second, we calculate the price $p(w)$, which is such that if each voter in $S(w)$ paid $p(w)$ or all the money she has left, then the voters from $S(w)$ would altogether pay n. Finally, among the above, we determine the alternative w with a minimal value for $p(w)$; we reduce the budget of every voter in $S(w)$ by $p(w)$ (or to 0 if their current budget is less than $p(w)$); we set the decision on I_j to w, and we continue with the next round, until no further purchase can be made. It might happen that after this procedure, there are issues for which the decision has not been set. For these, in the second phase, we select an alternative arbitrarily.

Conditional Proportional Approval Voting (cPAV)

An outcome w of a conditional approval election in an instance $P = (I, N, B)$ gains a score of $\sum_{k=1}^{u_i(w)} \frac{1}{k}$ from every voter v_i who is satisfied with w with respect to $u_i(w)$ issues. The cPAV score of w is $\sum_{v_i \in N} \sum_{k=1}^{u_i(w)} \frac{1}{k}$, or, in words, it is the sum of the scores that it gains from all the voters of the electorate. The outcome that achieves the highest cPAV score is the winning one under cPAV. Note that the only difference between the unconditional definition of PAV [Jan16; Thi95] and the version we suggest here, comes from the way that a voter's satisfaction with respect to an issue is defined. Therefore, the rule works in exactly the same way regardless of whether we have conditional or unconditional ballots.

[Jan16] Janson (2016): Phragmén's and Thiele's election methods.
[Thi95] Thiele (1895): Om Flerfoldsvalg.

Conditional Method of Equal Shares (cMES)

The unconditional version of cMES was introduced by Peters and Skowron [PS20a] (and was originally being referred to as "Rule X"). In our more general setting, the rule consists of two phases, and the first one works in rounds. Initially, each voter is given a budget m, equal to the number of issues and all

[PS20a] Peters and Skowron (2020): Proportionality and the Limits of Welfarism.

issues are being considered as unfixed. Fixing an issue will cost a price of n. Unlike in the unconditional version of the rule, we also allow that several issues are fixed at the same time, in which case the price to be paid is n times the number of issues to be fixed. For any round t of the first phase and an issue $I_j \in I$ that has not been fixed yet, denote by $\Gamma^u(I_j)$ the set of all issues in $\cup_{v_i \in N} \Gamma^*_i(I_j)$ that remain unfixed until round t. For every such issue I_j, for every set $I' \subseteq \Gamma^u(I_j)$, and for every possible (sub)outcome w on the issues of I', we perform the following: First, we identify the set of voters $S(w)$ who have a positive remaining budget and are satisfied with respect to I_j under w; second, we calculate[1] the price $p(w)$, which is such that if each voter in $S(w)$ paid $p(w)$ or all the money she has left, then the voters from $S(w)$ would altogether pay $n \cdot |I'|$. Among the above, we determine the set of issues I' and the suboutcome w with a minimal value for $p(w)$; we reduce the budget of every voter in $S(w)$ by $p(w)$ (or to 0 if their current budget is less than $p(w)$); we set the decision on the issues of I' to w, and we continue with the next round, until no further purchase can be made. It might happen that after this procedure, there are issues for which the decision has not been set. For these, in the second phase, we select an alternative arbitrarily.

A natural case for elections with conditional ballots, that we pay special attention to, is when all the voters agree on the dependencies among issues, i.e., when they have the same dependency graph.[2] With a common graph, the execution of cMES becomes a bit simpler. In particular, for a yet unfixed issue $I_j \in I$, we only look at the single subset of the in-neighborhood of I_j, say I', that has not been fixed in the previous rounds. For this set I', we check all possible suboutcomes w to identify the voters who can get satisfied with respect to I_j, and continue in the same manner as in the description of cMES above.

4.2 Conditional Proportional Approval Voting

We begin our study by examining cPAV, the conditional version of Proportional Approval Voting; a voting rule that exhibits significant proportionality guarantees in the binary and unconditional case [SG22]. The main result of this section is the identification of a proportionality guarantee under a certain assumption (Theorem 4.2). Before delving into this, we present an example that establishes that, as in the unconditional setting, we should not have too high expectations in terms of the α-value that is achievable.

[SG22] Skowron and Górecki (2022): Proportional Public Decisions.

Example 4.2 Consider an instance $P = (I, N, B)$, with m issues, such that m is a multiple of $d^{\Delta+1}$ and furthermore, suppose that m can be written as $m = k(\Delta + 1)$, for an integer k. Assume also that the issues of I can be partitioned in k sets, namely $M_1, M_2, \dots, M_k$, of size $(\Delta + 1)$ each, so that issues from M_i are not dependent on issues from M_j for any voter, and for

[1] The calculation of $p(w)$ for a given w can be done easily, see e.g. the proof of Theorem 4.7.

[2] In some scenarios, this may even be enforced by the election organizer, either for uniformity reasons or when there are obvious enough dependencies among issues that apply to all voters.

any $i \neq j$. Let there also be $d^{\Delta+1}$ voters. We fix an $i \in [k]$ and we focus on a single set of issues M_i. We will make each issue of M_i dependent on all the remaining Δ issues of the group M_i, for all voters. This means that the global dependency graph is a disjoint union of cliques.

To describe the voters' preferences, note that there are exactly $d^{\Delta+1}$ possible outcomes for the issues of any single clique. We define the preferences so that for every possible outcome in each clique, there is exactly one voter who is satisfied with that outcome, with respect to all the $\Delta + 1$ issues. Furthermore this voter is dissatisfied in any other outcome with respect to all these issues. For instance, say we fix a suboutcome $x = (x_1, \dots, x_{\Delta+1})$ for a particular clique. Then, we will have exactly one voter whose ballot is $\{x_{-1} : x_1,\ x_{-2} : x_2,\ \dots\ , x_{-(\Delta+1)} : x_{(\Delta+1)}\}$, where x_{-i} denotes the tuple $(x_1, \dots, x_{i-1}, x_{i+1}, \dots x_{\Delta+1})$. Thus, observe that if x is indeed the final selection for the clique under consideration, the satisfaction of the corresponding voter, with respect to these issues, equals exactly $\Delta + 1$. We use the same construction of preferences for the issues of the remaining cliques.

Proposition 4.1 *There does not exist a voting rule that is* 1*-proportional.*

Proof. Consider the instance P of Example 4.2, where $r_V = m, \forall V \subseteq N$, pick an arbitrary voting rule and let w be its winning outcome in P. Since the voters do not agree on any issue under any outcome, there exists at least one voter v_i satisfied with at most $\frac{m}{n}$ issues. If v_i is satisfied with strictly less than $\frac{m}{d^{\Delta+1}}$ issues, then consider the set $V = \{v_i\}$. Proportionality requires that $u_i(w) \geq m\frac{|V|}{n} = \frac{m}{d^{\Delta+1}}$, which is a contradiction. Hence, it must be that all voters are satisfied with exactly $\frac{m}{d^{\Delta+1}}$ issues. If we focus now on $V = N$, it should hold that $u_i(w) > m - 1$ for all $v_i \in N$, which is not the case. □

In fact, Example 4.2 has further negative implications.

Remark 4.1 Consider a rule that is reasonably fair to the voters, in the sense that it does not treat any voter in a significantly different manner. Then in Example 4.2, every voter would have to be satisfied with respect to exactly $\frac{m}{d^{\Delta+1}}$ issues. But then, looking at $V = N$ would imply that the rule cannot be α-proportional, for any $\alpha > \frac{1}{d^{\Delta+1}}$.

Consequently, the best we could expect is a proportionality guarantee of $\alpha = \frac{1}{d^{\Delta+1}}$. We have not yet been able to obtain such a general result for all instances. The analysis of the PAV score in conditional elections is more challenging to handle because the satisfaction of a voter can change more abruptly when we alter the value of a single issue, due to the effect this may have on other issues. On the positive side, we can prove a guarantee by restricting voters' preferences.

It remains open to determine whether such restrictions could be relaxed or removed.

Assumption 4.1 *For every voter $v_i \in N$, any two issues in R_i are either located in different components or are at a distance of at least 4 from each other in the global dependency graph.*

In other words, the assumption states that if a voter can be satisfied with respect to some issue, she cannot be satisfied with respect to to any other "nearby" issue in the global dependency graph. Hence it intends to capture voters who express preferences for a rather limited number of issues per component.

The most instructive, and intuitively simplest, yet non-trivial, case in which the assumption holds, appears when voters express preferences only for one issue per connected component. As a first example, consider instances with a common dependency graph for all voters, consisting of a collection of in-stars, where the voters may only be satisfied with respect to the central issue of each star, dependent on the leaf-issues. Generalizing this, we can also allow voters having different dependency graphs, as long as, again, there is a central issue in each component of the global dependency graph that the voters care about, albeit expressing different dependencies for it. Furthermore each voter could specify a different issue that she cares about in each component, justified by the voter's specialization, expertise or prioritization.

Theorem 4.2 *Under Assumption 4.1, cPAV is α-proportional for $\alpha = \frac{1}{(1+\Delta^2)d^{\Delta+1}}$.*

Proof. Consider a conditional election instance (I, N, B) in which Assumption 4.1 holds, let G be its global dependency graph, and w its cPAV winning outcome. For any fixed issue I_j, our proof will be based on a counting argument, where we will examine all possible ways of changing the alternatives chosen for I_j and its neighbors in w. Let s_j be a (sub)outcome that specifies an alternative for every issue in $\Gamma^*(I_j)$. Let $\delta_w(s_j)$, be the difference of the cPAV score if we change every issue of $\Gamma^*(I_j)$ from its value in w to the value indicated by s_j, minus the cPAV score of w. We denote by $\mathcal{F}_j$ the set of all possible suboutcomes s_j such that $s_j(I') \neq w(I')$, for at least one issue $I' \in \Gamma^*(I_j)$, so that s_j does impose an outcome different from w. Given that $|\Gamma^*(I_j)| \leq \Delta + 1$, it holds that $|\mathcal{F}_j| \leq d^{\Delta+1} - 1$.

The main component of the proof is the estimation of the expression

$$\sum_{I_j \in I} \sum_{s_j \in \mathcal{F}_j} \delta_w(s_j),$$

denoted by $\mathcal{S}_w$ for simplicity. Before we proceed, recall also that if u_i is the number of issues that v_i is satisfied with under w, the voter contributes to the cPAV score the quantity $1 + \frac{1}{2} + \cdots + \frac{1}{u_i}$. At what follows we will use $\Gamma(\Gamma(I_j))$ to denote the set of issues that are at distance 2 from I_j in G. We consider now two cases.

First, we consider an issue I_k with respect to which voter v_i is satisfied under w. We will take into account all possible tuples s_j in the expression $\mathcal{S}_w$, that may affect the satisfaction of v_i for this issue. These are precisely the suboutcomes $s_k \in \mathcal{F}_k$, the suboutcomes $s_\ell \in \mathcal{F}_\ell$ that correspond to any issue $I_\ell \in \Gamma(I_k)$, and also the suboutcomes $s_\ell \in \mathcal{F}_\ell$ for any issue $I_\ell \in \Gamma(\Gamma(I_k))$. Let us denote by $\mathcal{F}'_k$ the set of all these suboutcomes. In worst case, any change of w by the values indicated by some $s \in \mathcal{F}'_k$ may cause voter v_i to become dissatisfied with respect to issue I_k. Observe also that by Assumption 4.1, the suboutcomes from $\mathcal{F}'_k$ cannot affect any other issues that voter v_i is satisfied with under w, in other words, v_i will remain satisfied with all the other $u_i - 1$ issues that she was satisfied with under w. Therefore, when we change w according to a suboutcome $s \in \mathcal{F}'_k$, either the voter v_i does not contribute anything to $\mathcal{S}_w$, or her contribution will be negative, and equal to $-1/u_i$ due to the definition of cPAV. To bound the number of tuples that can create this negative contribution, we need to estimate $|\mathcal{F}'_k|$. To do this, note that the set $\Gamma^*(I_k) \cup \Gamma(\Gamma(I_k))$ contains exactly I_k itself, its neighbors, and its neighbors of neighbors which altogether correspond to at most $(1 + \Delta + \Delta(\Delta - 1)) = 1 + \Delta^2$ issues. Given also that for every such issue I_j, the number of different tuples $s_j \in \mathcal{F}_j$ to consider is $d^{\Delta+1} - 1$, it is true that

$$|\mathcal{F}'_k| \leq (1 + \Delta^2)(d^{\Delta+1} - 1).$$

Thus, the total contribution of voter v_i is in worst case at least

$$-\frac{(1 + \Delta^2)(d^{\Delta+1} - 1)}{u_i}.$$

Since we know that under w, there are exactly u_i issues that v_i is satisfied with, we conclude that for every $i \in [n]$ the contribution of v_i to $\mathcal{S}_w$ is at most $-\frac{u_i(1+\Delta^2)(d^{\Delta+1}-1)}{u_i}$.

Second, consider the $|R_i| - u_i$ issues that v_i is dissatisfied under w, but for which we know that there exists an assignment of values that satisfy v_i (by definition of R_i). Fix any such issue, say I_k. We know that in $\mathcal{F}_k$ there exists at least one suboutcome, say s_k, that can make v_i satisfied with respect to I_k. By Assumption 4.1, we know that changing w according to s_k cannot affect the remaining issues that v_i is satisfied with under w. Hence, there is at least one suboutcome, s_k, for which voter v_i contributes a positive value to $\mathcal{S}_w$, which equals $\frac{1}{u_i+1}$. Therefore, for every $i \in [n]$ the contribution of v_i to $\mathcal{S}_w$ is at least $\frac{|R_i|-u_i}{u_i+1}$.

The optimality of w implies that $\delta_w(s_j) \leq 0$, for every $s_j \in \mathcal{F}_j$ and for every $j \in [m]$. Furthermore, $-(d^{\Delta+1} - 1) \leq 0$ and, taking all the above into account, the following hold:

$$\begin{aligned}
0 \geq \sum_{I_j \in I} \sum_{s_j \in \mathcal{F}_j} \delta_w(s_j) \geq \\
\sum_{v_i \in N : u_i > 0} -\frac{u_i(1 + \Delta^2)(d^{\Delta+1} - 1)}{u_i} + \sum_{v_i \in N} \frac{|R_i| - u_i}{u_i + 1} \geq \\
\sum_{v_i \in N} \left(-(1 + \Delta^2)(d^{\Delta+1} - 1) + \frac{|R_i| - u_i}{u_i + 1} \right) = \\
\sum_{v_i \in N} \left(\frac{|R_i| + 1}{u_i + 1} - (1 + \Delta^2) d^{\Delta+1} \right)
\end{aligned}$$

Equivalently, $\sum_{v_i \in N} \frac{|R_i|+1}{u_i+1} \leq (1+\Delta^2)d^{\Delta+1}n$, and if we fix any set $V \subseteq N$, it holds that $\sum_{v_i \in V} \frac{|R_i|+1}{u_i+1} \leq (1+\Delta^2)d^{\Delta+1}n$. Using the fact that $|R_i| \geq r_V$, and by a rearrangement of the terms, we also have that

$$\sum_{v_i \in V} \frac{1}{u_i+1} \leq \frac{(1+\Delta^2)d^{\Delta+1}n}{r_V+1}. \tag{4.1}$$

Due to the harmonic and arithmetic mean inequality we get

$$\sum_{v_i \in V} \frac{1}{u_i+1} \geq \frac{|V|^2}{\sum_{v_i \in V}(u_i+1)} = \frac{|V|^2}{|V| + \sum_{v_i \in V} u_i}. \tag{4.2}$$

Harmonic and arithmetic mean inequality:

$$\frac{n}{\sum_{i \in [n]} \frac{1}{x_i}} \leq \frac{1}{n} \sum_{i \in [n]} x_i$$

Combining the relations (4.1) and (4.2) gives (after reordering)

$$\frac{1}{|V|} \sum_{v_i \in V} u_i \geq \frac{|V|}{n} \frac{r_V+1}{(1+\Delta^2)d^{\Delta+1}} - 1 > \frac{1}{(1+\Delta^2)d^{\Delta+1}} \frac{|V|}{n} r_V - 1.$$

Consequently, for every $V \subseteq N$, there exists a voter $v_i \in V$, for which $u_i(w) > \frac{1}{(1+\Delta^2)d^{\Delta+1}} \frac{|V|}{n} r_V - 1$. □

Despite the negative results indicated by Example 4.2, Theorem 4.2 provides the first guarantee of proportionality for conditional approval elections. On the downside, it is well-known that even in the unconditional setting, determining the winning outcome for PAV is NP-hard.

Implications for the Unconditional Case.

Assumption 4.1 trivially holds if all voters submit unconditional ballots since then, any two issues belong to different components in the global dependency graph (which does not have any edges). For this case, $\Delta = 0$ and the α-value achieved by Theorem 4.2 is $\frac{1}{d}$. This strictly generalizes the result of Skowron and Górecki [SG22], that deals only with the case of $d = 2$, and had left as an open problem the cases with higher values of d. Example 4.2 still works for $\Delta = 0$, making the result of Theorem 4.2 tight in the unconditional case, for any domain size, generalizing once again the analogous result.

Assumption 4.1: For every voter $v_i \in N$, any two issues in R_i are either located in different components or are at a distance of at least 4 from each other in the global dependency graph.

Theorem 4.2: Under Assumption 4.1, cPAV is α-proportional for $\alpha = \frac{1}{(1+\Delta^2)d^{\Delta+1}}$.

4.3 Conditional Method of Equal Shares

The computational intractability of PAV in the unconditional setting motivated the study of other rules, such as the Method of Equal Shares (MES), that overcome computational barriers and at the same time have desirable characteristics from the perspective of proportionality. In this section, we extend this line of work to conditional elections, by focusing on cMES.

4.3.1 Computational Complexity

The main result of this part of the dissertation is that Conditional MES is not, in general, computable in polynomial time, which is a noteworthy characteristic that differentiates it from its unconditional variant. This holds even for binary domains and rather simple dependency graphs, when either the maximum in-degree is large (Theorem 4.3) or the dependency graphs of the voters do not coincide (Theorem 4.5). Despite these negative results, there are well-motivated restricted families of instances for which we can compute cMES in polynomial time. In particular, when neither of the aforementioned conditions hold, a winning outcome can be computed efficiently (Theorem 4.7). Finally, for the case of different dependency graphs, we have also identified a restriction that implies polynomial-time computability (Theorem 4.8).

Theorem 4.3 *The winning outcome under cMES cannot be computed in polynomial time, unless P=NP, even when the voters have common dependencies.*

Proof. It is first necessary to discuss in more depth the input representation of instances, before delving into the details of the proof. Given an instance $P = (I, N, B)$, so far, we have considered that ballots of a voter $v_i \in N$ for an issue $I_j \in I$ are submitted in the form $\{s : t\}$ where s is a selection of alternatives for $\Gamma_i(I_j)$ and $t \subseteq D_j$. Hence, the voter can simply provide a list of all such ballots to indicate all the possible ways that she can be satisfied with respect to I_j (essentially a truth table). This is indeed computationally feasible when the maximum in-degree in a voter's graph is a constant, since in that case, a voter can provide a maximum of $d^{O(1)}$ ballots for any issue. We consider this to be the default way of describing the preferences when the maximum in-degree is sufficiently small. In cases where the in-degree is non-constant, this results in an exponential blow-up, and one needs alternative ways of describing the election instance. When this is the case, we can consider that ballots for an issue I_j are specified in the form $\{\tilde{s} : t\}$, where $t \subseteq D_j$ and $\tilde{s}$ is a succinct boolean formula when the domain is binary, or, for more general domains, a collection of constraints (like in the description of a constraint satisfaction problem (CSP)). In that case, a voter is satisfied with respect to I_j if we choose for $\Gamma_i(I_j)$ any combination of values that satisfies $\tilde{s}$ (which is expected to be satisfiable, otherwise the voter does not really have any dependence on the in-neighbors of I_j), and I_j is assigned a value from t.

To prove the statement we will reduce from the SAT variant in which the input has no duplicate clauses and has no variables that don't appear in any clause. Given such a SAT formula ϕ, on q variables, namely $x_1, x_2, \dots, x_q$, and r clauses, namely $c_1, c_2, \dots, c_r$, we create a conditional election instance $P = (I, N, B)$ of $|I| = q + 1$ issues and $|N| = r$ voters, as follows:

- For every $j \in [q]$, i.e. for every variable x_j of ϕ, we add an issue I_j in P. Moreover, we add a special issue I_0. Let $D_j = \{t_j, f_j\}, j \in [q]$ and $D_0 = \{\texttt{pos}, \texttt{neg}\}$.
- For every $i \in [r]$, i.e. for every clause c_i of ϕ, we add a voter v_i in P, who cares only for issue I_0 and votes for $\{c_i \wedge \bigwedge_{j:x_j \notin X(c_i)} (t_j \vee f_j) : \texttt{pos}\}$, where $X(c_i)$ is the set of variables that appear in the clause c_i.

Before moving on, we note that all voters have the same dependency graph, which is a star centered at I_0, and the in-degree in every voter's dependency graph is q. Moreover, every ballot is expressible in a succinct way, using a CNF formula, and hence, the construction is clearly polynomial. Most importantly, we highlight that if the SAT formula ϕ is satisfiable, then one can deduce an assignment for every issue $I_j, j \in [q]$, which, together with the option pos for I_0, satisfies all voters with respect to I_0, and vice versa. Equivalently, the following claim holds.

Claim 4.4 *The formula ϕ is satisfiable if and only if there exists an outcome to all issues of I that satisfies all voters with respect to issue I_0.*

In the remaining part of the proof we will use Claim 4.4 to show that, if we could efficiently determine the set of voters who should pay for the first purchase in a run of cMES, then we would also decide the satisfiability of ϕ, and vice versa. Since cMES is trying to identify in each iteration the minimum per voter cost among the possible purchases, the relevant decision version is to determine if the minimum per voter cost, between all possible outcomes that can be bought during the first round of cMES in P, is upper bounded by some given parameter, which in our case, will be set to $q + 1$.

During the first round of the execution of cMES, every voter is interested in buying a set of exactly $q + 1$ alternatives, otherwise she cannot be satisfied with respect to I_0, and furthermore, by construction, she can't be satisfied with respect to any other issue. Therefore, during its first iteration, cMES will search for the set of voters, who are interested in buying an alternative for all issues of I, that minimizes the per voter cost. For identifying such an outcome w, we note that $p(w)$ gets smaller as long as the size of $S(w)$ who participate in the payment, gets larger. More precisely, the issues of I cost in total $r(q + 1)$, and the per voter cost would be $p(w) = \frac{r(q+1)}{S(w)}$, where $|S(w)| \leq r$. Hence, it is obvious that the per voter cost in the first round of cMES is at most $q + 1$, if and only if, there exists an outcome w such that $S(w) = r$, or in other words that satisfies all voters with respect to I_0. By Claim 4.4, this occurs if and only if the formula ϕ is satisfiable. □

In the proof of Theorem 4.3, we have used in an essential way the fact that the maximum in-degree of the common dependency graph is non-constant. However, we expect that this represents rather extreme cases, and that in practical scenarios for conditional voting, it is more important to focus on the case of bounded in-degree, which has also been the focus of previous works on computational aspects of such elections, see e.g. [MP21b]. Even in this case, the NP-hardness remains, when the voters have different dependency graphs.

[MP21b] Markakis and Papasotiropoulos (2021): Winner Determination and Strategic Control in Conditional Approval Voting.

Theorem 4.5 *Breaking ties in favor of the largest set of buyers, the winning outcome under cMES cannot be computed in polynomial time, unless P=NP, even with a constant maximum in-degree in each voter's dependency graph.*

Proof. To prove the statement, we reduce from 3SAT. Given a 3SAT instance, Π, on q variables, namely $x_1, x_2, \dots, x_q$, and r clauses, namely $c_1, c_2, \dots, c_r$, we create a conditional election instance $P = (I, N, B)$ of $|I| = q + r + 1$ issues and $|N| = q + r + 1$ voters, as follows:

3SAT: Given a set of r clauses on 3 variables each, determine whether there exists an assignment to the variables that satisfies all clauses.

- We create a binary issue I_0 of domain $\{z_1, z_2\}$ and we add a voter voting unconditionally for z_1.
- For every $j \in [q]$, i.e. for every variable x_j of Π, we add a binary issue I_j of domain $\{t_j, f_j\}$. We refer to these issues as variable-issues. Furthermore, we add a variable-voter for every $j \in [q]$, who is only voting for $\{t_j : z_1, f_j : z_1\}$ and is dissatisfied with any other issue.
- For every clause c_j of Π, $j \in [r]$, we add a binary issue I'_j of domain $\{\texttt{pos}_j, \texttt{neg}_j\}$. We refer to these as clause-issues. Furthermore, we add a clause-voter who only cares to be satisfied with respect to I_0, and is voting for $\{c_j \wedge \texttt{pos}_j : z_1\}$, where c_j contains at most 3 variable-issues.

Observe that the maximum in-degree in every voter's dependency graph is at most 4. Furthermore, every voter is only interested in getting satisfied with respect to a single issue I_0.

Lemma 4.6 *The instance Π is a* YES*-instance if and only if there exists an outcome that satisfies all voters with respect to I_0.*

Proof of Lemma 4.6. For the forward direction, say that Π is a YES-instance of 3SAT. We will prove that in this case, we can indeed satisfy all the voters. Since Π is a YES-instance, there exists a boolean assignment x to the variables, that satisfies all clauses. Consider now the outcome of P that sets the variable-issues to the alternatives indicated by x, every clause-issue I'_j to $\texttt{pos}_j$ and I_0 to z_1. It is easy to verify that all voters of N are satisfied with this outcome.

Concerning the reverse direction, suppose that there is a choice for all the alternatives so that all voters are satisfied, each one with respect to I_0. Then, it also holds that the assignment for I_0 is z_1, that for every clause-issue I'_j the alternative $\texttt{pos}_j$ has been selected, and that the assignments to variable-issues satisfy all clauses c_j, or otherwise at least one voter would not have been satisfied. Consequently, there exists an assignment to every variable that satisfies all clauses, which means that Π is a YES-instance of 3SAT. ☒

In the remaining proof we will show that one cannot efficiently determine the set of voters who should pay for the first purchase in a run of cMES, unless P=NP. To do this, we establish the following claim: the set of all voters N, is the largest in cardinality set of voters that can jointly buy a set of alternatives at a minimum per voter cost in the first iteration if and only if the 3SAT formula is satisfiable.

For the forward direction, suppose that the formula is satisfiable. Then by Lemma 4.6, there is an outcome that satisfies all voters. This implies that the voters can buy an outcome of all the $q + r + 1$ issues, and the per voter cost would be $p(w) = q + r + 1$. If this was not the minimum possible per voter cost in the first iteration of cMES, then there was a different purchase, say for a suboutcome w', with $p(w) < q + r + 1$, due to the tie-breaking rule. We need to check whether there exists such a set of κ voters, that are willing to buy a suboutcome w' on λ issues, where $\kappa < q + r + 1$, and such that $p(w') < q + r + 1$. But then, with λ issues, we have $p(w') = \frac{\lambda(q+r+1)}{\kappa}$ and $\frac{\lambda(q+r+1)}{\kappa} < q + r + 1$ if and only if $\lambda < \kappa$. However, it must be the case that $\lambda \geq \kappa$. To see why, consider an arbitrary set of voters S that includes a variable-voters, b clause-voters and possibly the voter who votes for I_0 unconditionally. In order for such a set of voters to buy a suboutcome, note that the issue I_0 has to be included, and furthermore, each variable-voter requires her own variable issue to be included as well. For every clause-voter, we also have to include the corresponding clause issue, hence in total the set of issues that will be fixed is at least $a + b + 1$. Thus, it is impossible that κ voters can decide to buy together a suboutcome on less than κ issues.

For the reverse direction, suppose that in the first iteration of cMES, the set selected to buy alternatives is the set of all voters. Since, the ballot of every voter contains a distinct issue, this means that the voters have bought an alternative for all the issues in order to be concurrently satisfied with I_0. Then, by Lemma 4.6, the SAT formula is satisfiable. Hence, we have shown that we can run efficiently the first iteration of cMES in the instance we constructed, if and only if we can decide if the SAT formula is satisfiable. □

Moving to positive results, we demonstrate below that for a common dependency graph and bounded in-degree, cMES can be implemented efficiently. We find this to be a natural case (which complements the tractability landscape for cMES with Theorems 4.3 and 4.5), since there are numerous scenarios where the voters are likely to have the same perception on the dependence structure.

Theorem 4.3: The winning outcome under cMES cannot be computed in polynomial time, unless P=NP, even when the voters have a common dependency graph.

Theorem 4.5 Breaking ties in favor of the largest set of buyers, the winning outcome under cMES cannot be computed in polynomial time, unless P=NP, even with a constant maximum in-degree in each voter's dependency graph.

Theorem 4.7 *If all the voters have the same dependency graph, and* $\Delta_{\texttt{in}}$ *is bounded by a constant, then the winning outcome of cMES can be computed in polynomial time.*

Proof. Consider an instance $P = (I, N, B)$ of conditional approval elections such that the voters of N have the same dependency graph, called G. Let $\Delta_{\texttt{in}}$ be the maximum in-degree in G and say that $\Delta_{\texttt{in}} \in O(1)$. For any issue I_j, let also, for simplicity, $\Gamma^*(I_j)$ be the set of issues that contains I_j as well as the common set of in-neighbors of I_j for all voters. At any given round t of cMES, and for any unfixed issue $I_j \in I$, we say that $\Gamma^u(I_j)$ is the subset of the yet unfixed issues of $\Gamma^*(I_j)$. We need to check for each such I_j, all the possible assignments w to $\Gamma^u(I_j)$ and see if there is a purchase to be made in the current round.

We describe first how to do this for a particular w. Initially, we identify the set $S(w)$ of voters who are satisfied with respect to I_j, when we fix the issues of $\Gamma^u(I_j)$ as indicated by w. This can be done by checking all the ballots of the voters which are polynomially many. We then need to see if these voters can afford to buy the alternatives of w. If the total budget of the voters in $S(w)$ is less than $n \cdot |\Gamma^u(I_j)|$, then clearly w cannot be purchased. Otherwise, w is affordable, and we identify the price $p(w)$ such that if each voter v_i in $S(w)$ pays $p(w)$ or her remaining budget (if it is less than $p(w)$), then the total amount of payments equals $n \cdot |\Gamma^u(I_j)|$. Namely, if b_i is the current remaining budget of any voter v_i, then we want to identify the price $p(w)$ such that

$$\sum_{v_i \in S(w)} \min\{b_i, p(w)\} = n \cdot |\Gamma^u(I_j)|.$$

In order to determine $p(w)$, we can just start with setting $p(w) = \frac{n \cdot |\Gamma^u(I_j)|}{|S(w)|}$. If this cannot be afforded by every $v_i \in S(w)$, then we let the voters who cannot afford it pay their total remaining budget, and we compute the new average price for the remaining amount and among the remaining voters. We can continue in the same manner, and this process will terminate with an appropriate value for $p(w)$ (because we are in the case where $\sum_{v_i \in S(w)} b_i \geq n \cdot |\Gamma^u(I_j)|$). The process of finding $p(w)$ takes at most $O(n)$ steps.

For every issue I_j, there are at most $d^{\Delta_{\texttt{in}}+1}$ suboutcomes to check. This quantity is $d^{O(1)}$, since $\Delta_{\texttt{in}}$ is constant. Therefore, at each round, we have in total $m \cdot d^{O(1)}$ possible suboutcomes to check. Out of these, we need to find the one, called w, with the minimum price $p(w)$ (if any of them was affordable), which can also be executed in polynomial time as previously described. Then, we fix some issues as determined by w and continue to the next round. □

We conclude the computational analysis with one more positive result, applicable to the general setting of not necessarily common dependency graphs.

Theorem 4.8 *If each connected component of the global dependency graph has no more than a constant number of vertices, then the winning outcome under cMES can be computed in polynomial time.*

Proof. Say that each connected component of the global dependency graph is of size at most $c \in O(1)$. At a round t, cMES needs to consider all currently unfixed issues I_j and all subsets $I' \subseteq \Gamma^u(I_j)$, and examine every possible outcome $w \in \times_{I_k \in I'} D_k$. Obviously for any such I', all issues of I' belong to the same component in the global dependency graph. Hence, it holds that $|I'| \leq c$ and thus one needs to check no more that d^c possible ways to fix I_j together with a subset of its neighbors in the global dependency graph. There are at most $O(2^c)$ possible choices for the set I' and therefore, in every round, the identification of the minimum $p(w)$, for all possible (sub)outcomes $w \in \times_{I_k \in I'} D_k$, requires to check at most $O(m(2d)^c)$ choices. To conclude, the number of unfixed issues reduces between rounds until either all issues are fixed or no purchase can be made, thus, the procedure terminates after at most $O(m^2(2d)^c)$ steps, all of which can be executed in polynomial time. □

4.3.2 Proportionality Considerations

We first prove that in the conditional case, there is a family of instances for which cMES is strictly worse in terms of proportionality than cPAV.

Proposition 4.9 *If* $\Delta \geq 1$, *for any* $\rho \in \mathbb{R}_{\geq 1}$, *cMES is not* $\frac{1}{\rho}$*-proportional, even for instances that satisfy Assumption 4.1.*

Proof. We only need to consider the case of $\rho > 1$, since the case of $\rho = 1$ can be handled by Proposition 4.1. Furthermore, it is with out loss of generality to assume that $\rho \in \mathbb{N}_{>1}$. For this, we create an instance with $n = 2$ voters, namely v_1, v_2 and $m = 4\rho(\Delta + 1)$ issues, each of domain d. Say that the issues are partitioned in 4ρ sets and we focus on the first such, which say that includes the issues $I_1, I_2, \dots, I_{\Delta+1}$. Also say that the alternatives of issue I_j are $\{a_j^1, \dots, a_j^d\}$, for $j \in [\Delta + 1]$. Concerning the preferences of voter v_1 for these issues, she only has an unconditional ballot for I_1 that is $\{a_1^2\}$, i.e., she only approves the second alternative of this issue. On the other hand, v_2 also cares only for I_1 but her preference for this is dependent on all the remaining issues of the set, and more precisely, she is voting for $\{a_2^1 a_3^1 \cdots a_{\Delta+1}^1 : a_1^1\}$. The voters' preferences for the issues of the remaining sets are analogous.

The dependency graph of voter v_1 has no edges, the dependency graph of v_2 is a collection of directed stars, the global dependency graph consists of 4ρ connected components where each of them is an undirected star of size $(\Delta + 1)$ and the instance satisfies Assumption 4.1.

Assumption 4.1: For every voter $v_i \in N$, any two issues in R_i are either located in different components or are at a distance of at least 4 from each other in the global dependency graph.

Both voters have an initial budget of $4\rho(\Delta + 1)$, buying an issue costs 2, whereas buying a set of $(\Delta+1)$ issues of the same component costs $2(\Delta+1)$. Thus, it is cheaper for voter v_1 to buy any single alternative than for voter v_2 to buy any set of $(\Delta + 1)$ alternatives, and there are no other options during the execution of cMES for buying alternatives, given that the voters cannot get jointly satisfied for any single issue. Therefore, as long as voter v_1 has enough money left, she can buy any alternative that she likes. We observe that v_1 can be satisfied with respect to at most 4ρ issues and in order to attain this, she needs to pay (sequentially) 8ρ which is no more than her initial budget, for $\Delta \geq 1$. After these purchases, v_1 is not interested in buying any other alternative and at the same time v_2 can't be satisfied with respect to any issue, given the alternatives that are already fixed. This results to an outcome w for which $u_2(w) = 0$. Suppose that cMES is $\frac{1}{\rho}$-proportional. Then, for the set $V = \{v_2\}$, it is true that $r_V = 4\rho$ and hence it should hold that $u_2(w) > \frac{1}{\rho}4\rho\frac{1}{2} - 1$, which is a contradiction. □

Therefore, restrictions of a different flavor than Assumption 4.1 are necessary in order to end up with a bounded proportionality guarantee for cMES. To understand when to expect a good behavior from cMES, it is important to revisit first the unconditional case, where an assumption was also needed to achieve any proportionality bound. More precisely, the guarantee for MES by Skowron and Górecki [SG22] was established under the assumption that there are no abstainers, i.e., every voter approves at least one alternative from every issue, hence $r_V = m$. In the conditional setting, we will use a generalization of this statement, in the following form: no matter how the in-neighbors of an issue are set, a voter can still be satisfied with at least one alternative of that issue.

[SG22] Skowron and Górecki (2022): Proportional Public Decisions.

Assumption 4.2 *For every issue I_j and for every voter v_i, we assume that for every combination of values for the issues in $\Gamma_i(I_j)$, there is a choice for I_j that satisfies v_i with respect to I_j. When $\Gamma_i(I_j) = \emptyset$, we simply assume that v_i approves at least one alternative from the domain of I_j.*

We note that Assumption 4.2 is incomparable to the assumptions needed for the polynomial algorithms of Section 4.3.1 and that it implies $r_V = m$, for any set of voters V.

The assumptions suggested in Section 4.3.1 had to do with the commonalities of voters' dependency graphs as well as with the maximum in-degree of these graphs.

Theorem 4.10 *Under Assumption 4.2, cMES is α-proportional for $\alpha = \frac{1}{(\Delta+1)d^{\Delta+1}}$. When all voters have the same dependency graph, then the same bound holds with Δ replaced by Δ_{in}, the maximum in-degree over all issues.*

Proof. For clarity in presentation, we will prove the statement for the case of a common dependency graph for all voters; the general case can be proven in a straightforward fashion.

Towards a contradiction, suppose that cMES is not $\frac{1}{(\Delta_{\texttt{in}}+1)d^{\Delta_{\texttt{in}}+1}}$-proportional. Hence, there exists a conditional approval election instance $P = (I, N, B)$, with a winning outcome under cMES, say w, that violates the desired proportionality guarantee. Let $V \subseteq N$ be a set of voters such that for every voter $v_i \in V$ it holds that

$$u_i(w) \leq \frac{1}{(\Delta_{\texttt{in}}+1)d^{\Delta_{\texttt{in}}+1}} m \frac{|V|}{n} - 1.$$

We introduce the following notation: for any set $N' \subseteq N$ we say that $b(N')$ is the total remaining budget of the voters in N' at the end of the first phase of cMES, and $b_t(N')$ is the total remaining budget of the voters in N' at the beginning of round t of the first phase. Furthermore, henceforth, we will refer to the, undirected, common dependency graph of the voters of P, as G.

Claim 4.11 *Consider a (sub)outcome w' that was purchased during some round of the first phase of cMES. Then, if $V \cap S(w') \neq \emptyset$, it holds that $p(w') \leq \frac{(\Delta_{in}+1)d^{\Delta_{in}+1}n}{|V|}$.*

We suppose that Claim 4.11 holds and we postpone its proof for later. Therefore, every $v_i \in V$ paid at most $\frac{(\Delta_{\texttt{in}}+1)d^{\Delta_{\texttt{in}}+1}n}{|V|}$ for any of her purchases and we know that if v_i paid for purchasing a (sub)outcome w' on issues $I' \subseteq I$, she is satisfied with respect to at least one issue of I'. Hence, voter v_i had paid in total no more than

$$u_i \cdot max\{p(w') : u_i \in S(w')\} \leq m - \frac{(\Delta_{\texttt{in}}+1)d^{\Delta_{\texttt{in}}+1}n}{|V|}.$$

Since the initial budget of every voter is m, for every voter $v_i \in V$ it holds that

$$b(v_i) \geq \frac{(\Delta_{\texttt{in}}+1)d^{\Delta_{\texttt{in}}+1}n}{|V|}. \tag{4.3}$$

Therefore, for the voters of V, we have that $b(v_i) > 0$, and hence, we know that during the second phase of cMES, at least one issue was set at random. Otherwise it would have to hold that the budget of every voter of N is down to zero.

We focus on such an unfixed issue I_j, at the end of the first phase. Let us denote by $\Gamma^u(I_j)$ the set consisting of I_j and of all its currently unfixed in-neighbors in G (possibly zero). By Assumption 4.2, we know that there is a fraction of at least $\frac{1}{d^{|\Gamma^u(I_j)|}}$ of the voters of V that agree on a common outcome that satisfies them with respect to I_j. Clearly we have that

$$|\Gamma^u(I_j)| \leq \Delta_{\mathtt{in}} + 1.$$

Hence we have a set $V' \subseteq V$, which is at least a $\frac{1}{d^{\Delta_{\mathtt{in}}+1}}$ fraction of V who agree on how to fix the set $\Gamma^u(I_j)$ so as to be satisfied with respect to I_j.

From Equation (4.3) we get that the currently available budget of V' satisfies

$$b(V') \geq \frac{(\Delta_{\mathtt{in}} + 1)d^{\Delta_{\mathtt{in}}+1}n}{d^{\Delta_{\mathtt{in}}+1}} = (\Delta_{\mathtt{in}} + 1)n,$$

which is enough to buy an outcome for the issues of $\Gamma^u(I_j)$. But this means that cMES should not have terminated its first phase, reaching a contradiction. To complete the proof, it only remains to prove Claim 4.11.

Proof of Claim 4.11. Towards a contradiction, suppose that at some round t of the first phase, a voter v_i of V participated in a purchase of a (sub)outcome w' that concerns a set of issues $I' \subseteq I$, such that

$$p(w') > \frac{(\Delta_{\mathtt{in}} + 1)d^{\Delta_{\mathtt{in}}+1}n}{|V|}.$$

If there have been multiple such purchases, we focus at the very first one. Hence, before round t, all voters of V were paying at most $\frac{(\Delta_{\mathtt{in}}+1)d^{\Delta_{\mathtt{in}}+1}n}{|V|}$ for each of their purchases. Their satisfaction before round t is at most equal to their final satisfaction and therefore, in analogy to Equation (4.3),

$$b_t(v_i) \geq \frac{(\Delta_{\mathtt{in}} + 1)d^{\Delta_{\mathtt{in}}+1}n}{|V|}, \forall v_i \in V.$$

We come back now to round t, where a (sub)outcome concerning I' was purchased. Due to the definition of cMES, there exists at least one issue I_j, and a set $\Gamma^u(I_j)$, formed by I_j and its in-neighbors in G that were still unfixed at round t, such that $\Gamma^u(I_j) \subseteq I'$. By Assumption 4.2 and by the pigeonhole principle, there exists a fraction of $\frac{1}{d^{|\Gamma^u(I_j)|}}$ voters from V, called V'', who agree on a common outcome, say w'', for the issues in $\Gamma^u(I_j)$. Given that $b_t(v_i) \geq \frac{(\Delta_{\mathtt{in}}+1)d^{\Delta_{\mathtt{in}}+1}n}{|V|}$ for every $v_i \in V$, it also holds that

$$b_t(V'') \geq (\Delta_{\mathtt{in}} + 1)d^{\Delta_{\mathtt{in}}+1-|\Gamma^u(I_j)|}n,$$

which is greater or equal to $|\Gamma^u(I_j)|n$, since $\Delta_{\mathtt{in}} + 1 \geq |\Gamma^u(I_j)|$. Therefore, the voters from V'' can buy the issues of $\Gamma^u(I_j)$ by satisfying

$$p(w'') \leq \frac{(\Delta_{\mathtt{in}} + 1)d^{\Delta_{\mathtt{in}}+1}n}{|V|}.$$

Thus, cMES should have fixed $\Gamma^u(I_j)$ instead of I' by charging each voter that is willing to pay for w'' no more than $p(w'')$, which contradicts our initial assumption that at round t cMES made some voter from V pay more than $\frac{(\Delta_{\texttt{in}}+1)d^{\Delta_{\texttt{in}}+1}n}{|V|}$. ⊠

This completes the proof of Theorem 4.10. □

For the case of a common dependency graph for all voters, we show that we can also go a small step further by slightly relaxing Assumption 4.2, in a way that Theorem 4.10 still holds and at the same time no significant improvements on the proportionality bound would be possible. Before that, we give the following notation: consider an instance $P = (I, V, B)$, the voters of which have the same dependency graph. Then for every issue $I_j \in I$, let $\Gamma_{\texttt{in}}(I_j)$ be the set of in-neighbors of issue I_j in the dependency graph and let $Z(I_j) = \{I_k \in I \text{ such that } I_k \in \Gamma_{\texttt{in}}(I_j) \wedge I_j \in \Gamma_{\texttt{in}}(I_k)\}$.

Assumption 4.2 For every issue I_j and for every voter v_i, we assume that for every combination of values for the issues in $\Gamma_i(I_j)$, there is a choice for I_j that satisfies v_i with respect to I_j. When $\Gamma_i(I_j) = \emptyset$, we simply assume that v_i approves at least one alternative from the domain of I_j.

Assumption 4.3 *For every issue I_j and for every voter v_i, we assume that for every combination of values for the issues in $\Gamma_{in}(I_j) \setminus Z(I_j)$, there is a choice of values for the issues in $Z(I_j) \cup \{I_j\}$ that satisfies v_i with respect to I_j. When $\Gamma_{in}(I_j) = \emptyset$, we simply assume that v_i approves at least one value from the domain of I_j.*

It is easy to verify[3] that, for the case of a common dependency graph, Theorem 4.10 still works under this weaker assumption. Furthermore, we exhibit below that under Assumption 4.3, a proportionality guarantee that is significantly better than the result of Theorem 4.10 is impossible.

Proposition 4.12 *For $\Delta_{in} \geq 1$ and $d \geq 2$, cMES is not $\frac{1}{d^{\Delta_{in}+1}}$-proportional, even for instances that satisfy Assumption 4.3 and even if all voters have the same dependency graph.*

Proof. Towards a contradiction we create the instance $P = (I, N, B)$, with $d = 2$ and $\Delta_{\texttt{in}} = 1$. Say that $|I| = 6$ and $|N| = 4$ and that the domain of issue $I_j \in I$ is $\{x_j, y_j\}$, for $j \in \{1, 2, \ldots, 6\}$. Also say that b_i is the ballot of voter $v_i \in N$ and let the ballots be as follows:

$$b_1 = \{x_1 : x_2,\ x_2 : x_1,\ x_3 : x_4,\ x_4 : x_3,\ x_5 : x_6,\ x_6 : x_5\}$$
$$b_2 = \{x_1 : y_2,\ y_2 : x_1,\ x_3 : y_4,\ y_4 : x_3,\ x_5 : y_6,\ y_6 : x_5\}$$
$$b_3 = \{y_1 : x_2,\ x_2 : y_1,\ y_3 : x_4,\ x_4 : y_3,\ y_5 : x_6,\ x_6 : y_5\}$$
$$b_4 = \{y_1 : y_2,\ y_2 : y_1,\ y_3 : y_4,\ y_4 : y_3,\ y_5 : y_6,\ y_6 : y_5\}$$

The dependency graphs of the voters coincide, the global dependency graph of the instance consists of 3 independent edges and for any $I_j \in I$ there are no issues in $\Gamma_{\texttt{in}}(I_j) \setminus Z(I_j)$; hence Assumption 4.3 holds.

[3] The reason is that the issues in $Z(I_j)$ cannot be fixed before the round where I_j gets fixed, as they depend on it. Hence, in the proof of Theorem 4.10, we only need to impose the condition of Assumption 4.2 for the issues in $\Gamma_{\texttt{in}}(I_j) \setminus Z(I_j)$ (the ones that might have been fixed before the round where I_j is bought).

All voters initially have a budget of $|I| = 6$, but buying a pair of issues costs $2|N| = 8$. At the same time, they are not interested in buying a single issue. Most importantly, no two voters agree on purchasing together a common outcome for any pair of issues. Therefore, during the first phase of cMES, no issue will be fixed, and the mechanism will just select an outcome at random during the second phase. Without loss of generality, say that the outcome $(x_1, x_2, \dots, x_6)$ is selected. Then all voters other than the first one have a satisfaction score of 0. If cMES was $\frac{1}{d^{\Delta_{\mathtt{in}}+1}}$-proportional, then for the group V that consists of all voters except the first one, there should exist a voter of satisfaction strictly greater than $\frac{1}{2^2} 6 \frac{3}{4} - 1 = 0.125$, a contradiction. □

Implications for the Unconditional Case.

In the unconditional case, cMES coincides with MES and is computable in polynomial time. A corollary of Theorem 4.10 that concerns the unconditional case ($\Delta = 0$) is the generalization of the proportionality guarantee for MES from the case of binary decisions to any domain size d (which was left as an open question by Skowron and Górecki [SG22]), while meeting the lower bound of $\frac{1}{d}$ from Example 4.2.

Theorem 4.10: Under Assumption 4.2, cMES is α-proportional for $\alpha = \frac{1}{(\Delta+1) d^{\Delta+1}}$.

4.4 Concluding Discussion and Future Directions

We studied generalizations of two well-known voting rules with proportionality guarantees, PAV and MES, to the setting of conditional approval elections, in which voters' exhibit preferential dependencies between the issues under consideration. Our main results establish that both cPAV and cMES can achieve proportionality bounds, under different assumptions. Conditional Proportional Approval Voting seems to favor situations where the satisfaction score of a voter is somewhat restricted, whereas Conditional Method of Equal Shares has a better behavior when voters are "easier" to please for every issue.

There are several questions for potential future work. We first note that we do not have yet a complete picture about the tightness of our bounds. Also it has been challenging to understand whether the assumptions used can be relaxed, and to what extent. The assumption on cMES seems to be quite critical even in its unconditional variant; as for cPAV, we are optimistic that relaxations might be possible, given also the fact that under unconditional ballots, cPAV works well without any assumptions. One can also study the behavior of other rules under the conditional setting, or think of further ways to generalize cMES, based on how a purchase is made and who participates in each purchase. In general, we believe that proportional representation in combinatorial domains is a fascinating area, worth further exploration.

Part Two
Delegative Democracy

An Approval-Based Model for Liquid Democracy 5

Liquid Democracy (LD) is a voting paradigm that has emerged as a flexible model for enhancing engagement in decision-making. The main idea in LD models is that a voter can choose either to vote herself or to delegate to another voter that she trusts to be more knowledgeable or reliable on the topic under consideration. A delegation under LD, is a transitive action, meaning that the voter not only transfers her own vote but also the voting power that has been delegated to her by others. Experimentations and real deployments have already taken place using platforms that support decision-making under Liquid Democracy. One of the first such systems that was put to real use was Župa, intended for a student union for the University of Novo Mesto in Slovenia. Another example is LiquidFeedback that was used by the German Pirate party (among others). Other political parties (such as the Flux Party in Australia) or regional organisations have also attempted to use or experiment with LD, leading to a growing practical appeal. Even further examples include the experiment run by Google via Google Votes, as well as Civicracy and, the more recently developed, Sovereign. We refer to [Pau20] for an informative survey on these systems.

[Pau20] Paulin (2020): An Overview of Ten Years of Liquid Democracy Research.

The interest generated by these attempts, has also led to theoretical studies on relevant voting models and has enriched the research agenda of the community. The goals of these works have been to provide more rigorous foundations and highlight the advantages and the negative aspects of LD models. Starting with the positive side, LD definitely has the potential to incentivize civic participation, both for expert voters on a certain topic, but also for users who feel less confident and can delegate to some other trusted voter. At the same time, it also forms a flexible means of participation, since there are no restrictions for physical presence, and usually there is also an option of instant recall of a delegation, whenever a voter no longer feels well represented.

Coming to the critique that has been made on Liquid Democracy, an issue that can become worrying is the formation of large delegation paths. Such paths tend to be undesirable since a voter who gets to cast a ballot may have a rather different opinion with the first voters of the path, who are being represented by her [Gre15]. Secondly, LD faces the risk of having users accumulating excessive voting power, if no control action is taken [BZ16]. Furthermore, another undesirable phenomenon is the creation of delegation cycles, which could result to a waste of participation for the involved voters. Despite the criticism, Liquid Democracy is still a young and promising field for promoting novel methods of participation and decision-making, generating an increasing interest in the community towards tackling some of the existing criticism but also identifying additional inherent problems.

[Gre15] Green-Armytage (2015): Direct Voting and Proxy Voting.

[BZ16] Blum and Zuber (2016): Liquid Democracy: Potentials, Problems, and Perspectives.

Contribution.

In the current chapter, we focus on a model where voters can have approval-based preferences on the available actions. Each voter can have a set of approved delegations, and may also approve voting herself or even abstaining. Our main goal is the study of centralized algorithms for optimizing the overall satisfaction of the voters. For this objective, under our model, it turns out that it suffices to focus only on delegations to actual voters (i.e., delegation paths of unit length). Even with this simpler solution space, the problems we study turn out to be computationally hard.

In Section 5.2, we start with the natural problem of minimizing the number of dissatisfied voters, where we establish a connection with classic combinatorial optimization problems, such as SET COVER and DOMINATING SET. We present approximation preserving reductions which allow us to obtain almost tight approximability and hardness results. The main conclusion from these is that one can have a small constant factor approximation when each voter approves a small number of possible representatives. A constant factor approximation can also be obtained for the variant of maximizing the number of satisfied voters, through a different approach of modeling this as a constraint satisfaction problem.

Moving on, in Section 5.3, we consider the design of exact algorithms for the same problems. Our major highlight is the use of a logic-based technique, where it suffices to express properties by formulas in Monadic Second Order logic. In a nutshell, this approach yields an FPT algorithm, whenever the treewidth of an appropriate graph-theoretic representation of our problem is constant. Under the same restriction, polynomial time algorithms also exist when adding certain secondary objectives on top of minimizing (resp. maximizing) dissatisfaction (resp. satisfaction). To our knowledge, this framework has not received much attention in the Social Choice community and we expect that it could have further applicability for related problems.

Related Work.

To position the work presented in current chapter with respect to existing literature, we note that the works most related to ours are [GKM+21] and [EGP19]. In terms of the model, we are mostly based on [GKM+21], which studies centralized algorithms and where voters specify possible delegations in an approval format. Coming to the differences, their model does not allow abstainers (which we do), but more importantly, [GKM+21] studies a different objective and no notion of satisfaction needs to be introduced (in Section 5.3 we also examine a re-

[GKM+21] Gölz et al. (2021): The Fluid Mechanics of Liquid Democracy.

[EGP19] Escoffier et al. (2019): The Convergence of Iterative Delegations in Liquid Democracy in a Social Network.

lated question). Our main optimization criteria are inspired mostly by [EGP19], which among others, tries to quantify voters' dissatisfaction. Our differences with [EGP19] is that they have voters with rank-based preferences and their optimization is with respect to equilibrium profiles and not over all possible delegations (in Section 5.4, we also provide a game-theoretic direction with some initial findings). These works, like ours, are agnostic to the final election outcome (preferences are with respect to delegations and not on actual votes).

More generally, the LD-related literature within Computational Social Choice concerns *(i)* comparisons with direct democracy models, *(ii)* game-theoretic stability of delegations, *(iii)* axiomatic approaches. Concerning the first topic, local delegation mechanisms, under which every voter independently is making a choice, have been explored in [CM19; KMP21]. For the second direction, one can view an LD framework as a game in which the voters can make a choice according to some given preference profile. Such games have been considered in [EGP19] and [EGP20]. At the same time, game-theoretic aspects have also been studied in [BGL19] and, for the case of weighted delegations, in [ZG21]. Concerning the third direction, a range of delegation schemes have been proposed to avoid delegation cycles [KKM+20], accumulation of high power [GKM+21] and existence of inconsistent outcomes [CG17]. Further paradigms related to LD have also been considered, e.g. in [AM19; BBC+11; CGN21].

5.1 Election Framework and Definitions

5.1.1 Approval Single-Step Liquid Model

We denote by $V = \{1, \dots, n\}$ the set of agents who participate in the election process and we will refer to all members of V as voters (even though some of them may eventually not vote themselves). In the suggested model, which we refer to as Approval Single-Step Liquid model (ASSL), each voter $i \in V$ needs to express her preferences, in an approval-based format, on the options of

(i) casting a ballot herself,
(ii) abstaining from the election,
(iii) delegating her vote to voter $j \in V \setminus \{i\}$.

Namely, a voter may approve any combination of

➛ casting a ballot herself. We let C denote the set of all such voters.

➛ abstaining from the voting procedure (e.g., because she feels not well-informed on the topic). We let A denote the set of all such voters.

➛ delegating her vote to some other voter she trusts. For every $v \in V$, we denote by $N(v)$ the set of approved delegatees of v.

Note that we place no restriction on whether a voter accepts one or more of the above options (or even none of them). Hence, in a given instance it may be true that $C \cap A \neq \emptyset$ or that $C \cup A = \emptyset$ or that $v \in C$ and at the same time $N(v) \neq \emptyset$, etc. It is often natural and convenient to think of a graph-theoretic representation of the approved delegations. Hence, for every instance, we associate a directed graph $G = (V, E)$, such that $N(v)$ is the set of out-neighbors of v, i.e., $deg^+(v) = |N(v)|$, where $deg^+(v)$ is the out-degree of v. This will be particularly useful in Section 5.3.

Let a delegation function $d : V \to V \cup \{\perp\}$ express the final decision for each voter. We say that $d(v) = v$, if voter v votes, $d(v) = \perp$ if she abstains, and $d(v) = u \in N(v)$, if she delegates to voter u. Given a delegation function $d(\cdot)$, we refer to a voter who casts a ballot as a *guru*. The guru of a voter $v \in V$, denoted by $gu(v)$, can be found by following the successive delegations, as given by a delegation function $d(\cdot)$, starting from v until reaching a guru (if possible). Formally, $gu(v) = u$ if there exists a sequence of voters $u_1, \dots, u_\ell$ such that $d(u_k) = u_{k+1}$ for every $k \in \{1, 2, \dots, \ell - 1\}, u_1 = v, u_\ell = u$ and $d(u) = u$. Obviously, $gu(v) = v$ if v votes. In case the delegation path starting from v ends up in a voter u for which $d(u) = \perp$ then we say that v does not have a guru and we set $gu(v) = \infty$. Additionally, we do the same for the case where the successive delegations starting from some voter v form or end up in a cycle.

We say that a voter v is satisfied with the delegation function $d(\cdot)$ if v approves the outcome regarding her participation or her representation by another guru-voter. This means that either $d(v) = v$ and $v \in C$ or that $d(v) = \perp$ and $v \in A$ or that $d(v) = u$, with $u \neq v$, $u \neq \perp$ and $gu(u) \in N(v)$. In all other cases, the voter is dissatisfied. Our work mainly deals with the problem of finding centralized mechanisms for the following computational problems:

MINIMUM SOCIAL COST (MIN-SC)/MAXIMUM SOCIAL GOOD (MAX-SG)	
Given:	An instance of ASSL, i.e., the approval preferences of n voters regarding their intention to vote, abstain and delegate.
Output:	A delegation function that minimizes the number of dissatisfied voters/maximizes the number of satisfied voters.

5.1.2 Warm-up Observations

We start with some observations that will help us tackle the algorithmic problems under consideration. Given an instance of ASSL, let G be the corresponding graph with the approved delegations, as described in the previous subsection. A delegation function $d(\cdot)$, induces a subgraph of G that we denote by $G(d)$, so that (u, v) is an edge in $G(d)$ if and only if $d(u) = v$. Clearly, the out-degree of every vertex in $G(d)$ is at most one and thus it can contain isolated vertices, directed trees oriented towards the gurus, but in general it can also contain cycles, the presence of which can only deteriorate the solution. The next proposition shows that we can significantly reduce our solution space.

Proposition 5.1 *Consider a solution given by a delegation function* $d(\cdot)$*. There always exists a solution* $d'(\cdot)$ *which is at least as good (i.e., the number of satisfied voters is at least as high) as* $d(\cdot)$*, so that* $G(d')$ *is a collection of disjoint directed (towards the central vertex) stars, and voters that abstain form isolated vertices.*

Proposition 5.1 is the one that justifies the name Approval Single-Step Liquid (ASSL) of the suggested model.

Proof. First of all say that $G(d)$ has a directed cycle. Consider the voters who are part of that cycle and delete all the edges of the cycle by having the corresponding voters to either vote or abstain (choose arbitrarily). All the voters in the cycle were dissatisfied before, since $gu(v) = \infty$ for each such v. After destroying the cycle, some of them may become satisfied. Also, anyone who was delegating to these voters was dissatisfied before, and now she either remains this way or becomes satisfied, so the new solution is no worse than the initial one.

Suppose now that $G(d)$ is not a collection of directed stars. Since there are no directed cycles, consider the longest path of delegations in $G(d)$ (with length at least 2). Let v be a source vertex of the path (one with in-degree 0), and let u be a sink vertex (with out-degree 0). There are 2 cases to consider.

Case 1: v is satisfied. This can happen if $d(u) = u$ and $u \in N(v)$. But by the assumptions of ASSL, this directly means that $(v, u) \in E$, and we can replace the edge starting from v in $G(d)$ with the edge (v, u). Voter v is still satisfied, and since v has in-degree 0 in $G(d)$, no other voter is affected by this change. Hence, we have a solution with the same number of satisfied voters.

Case 2: v is dissatisfied. This can happen either if $d(u) = u$ and v does not accept u as her guru of if $d(u) = \bot$. Then we can simply delete the edge starting from v and have v either vote herself or abstain (we can choose arbitrarily). The satisfaction of other voters is not affected, whereas v may now be satisfied so the change we made does not worsen the solution.

Thus, in all cases, we have produced a new solution with at least the same number of satisfactions, and where the length of the delegation path we considered has been reduced. By repeating this argument, we can reach a solution where $G(d')$ is simply a collection of directed stars, and especially for vertices who abstain, we can enforce that they will be isolated vertices. □

Remark 5.1 With Proposition 5.1 in mind, one may discern similarities with proxy voting models (see e.g. [AFV+21] or Chapter 7), under which every voter is being represented by her delegator, since no transitivity of votes is taken into account. Nevertheless, we still like to think of our model as a Liquid Democracy variant, because it is precisely the objectives that we study together with the centralized approach that enforce Proposition 5.1. When discussing decentralized scenarios or game-theoretic questions (as we do in Section 5.4), longer delegation paths may also appear.

The next proposition shows that for certain voters, we can a priori determine their action, when looking for an optimal solution and that we can be sure about the action of any voter who is dissatisfied under a given delegation function.

Proposition 5.2 *Consider a solution given by a delegation function $d(\cdot)$. There always exists a solution which is at least as good (i.e., the number of satisfied voters is at least as high) as $d(\cdot)$, in which (i) every voter in C casts a ballot and (ii) if any voter is dissatisfied, it is because she is casting a ballot without approving it.*

Proof. For the first part let $v \in C$ and suppose that $d(v) \neq v$. There are 2 cases to consider.

Case 1: $d(v) = \perp$. By Proposition 5.1, v is an isolated vertex in $G(d)$, so no other voter is affected by any change of his vote. Then, in case she is dissatisfied, we can improve the solution by having him cast a vote. If she is satisfied ($v \in A \cap C$), we can still maintain his satisfaction by having him vote.

Case 2: $d(v) = u$ for some $u \neq v$. If v is satisfied, she will remain satisfied by asking him to cast a vote. If she is dissatisfied, then we can make him satisfied and improve the solution. By Proposition 5.1, the in-degree of v is 0 in $G(d)$, therefore no other voter is affected by this change.

Thus, in both cases, we have produced a new solution with at least the same number of satisfactions as before.

Similarly, for the second part, if a voter v abstained or delegated to another voter and became dissatisfied, we could ask her to vote and this may only improve the solution since some of her in-neighbors may become satisfied whereas the satisfaction of v can't be worse than it was before. □

Proposition 5.2 takes care of voters in C. We cannot state something similar for the rest of the voters, since it might be socially better to dissatisfy a certain voter by asking her to cast a ballot so as to make other people (pointing to her) satisfied. In practice, this can also occur in cases where voting may be costly (in time or effort to become more informed) and one member of a community may need to act in favor of the common good, outweighing her cost.

5.2 Approximation Algorithms and Hardness Results

In this section we will mainly focus on MIN-SC, but we will also examine MAX-SG in Section 5.2.2 and further related questions in Section 5.2.3. We pay particular attention to instances where every voter approves only a constant number of other voters, i.e., $\Delta = \max_v |N(v)| \in O(1)$.

We find the case $\Delta \in O(1)$ to be a realistic one, as it is rather expected that voters cannot easily trust a big subset of the electorate.

5.2.1 Social Cost Minimization

We start by showing that the problem is intractable even when each voter approves at most 2 other voters. In fact, we show that our problem encodes a directed version of the DOMINATING SET problem, hence, beyond NP-hardness, we also inherit known results concerning hardness of approximation.

Theorem 5.3 *Let* $\Delta = \max_{v \in V} |N(v)| \geq 2$*. When* Δ *is constant, it is NP-hard to approximate* MIN-SC *with a ratio smaller than* $\max\{1.36, \Delta - 1\}$*. For general instances, it is NP-hard to achieve an approximation better than* $\ln n - \Theta(1) \ln \ln n$*.*

Proof. We will use an approximation preserving reduction from the problem DIRECTED MINIMUM DOMINATING SET (MIN-DS$_d$). Let $G = (V, E)$ be the input graph of an instance I of MIN-DS$_d$. We construct an instance I' of MIN-SC as follows: for every vertex $v \in V$ of instance I, we have a voter $\hat{v}$ in I' who approves precisely her in-neighbors in G. More formally, the approved set for $\hat{v}$ is $N(\hat{v}) = \{\hat{u} : (u, v) \in E\}$. Hence, the corresponding graph of instance I' is $G' = (V, E')$ where $E' = \{(u, v) : (v, u) \in E\}$. We also assume that the voters do not approve neither voting themselves nor abstaining, hence $C \cup A = \emptyset$. Note that if G has in-degree bounded by b, then G' has out-degree bounded by b.

MIN-DS$_d$: Given a directed graph $G = (V, E)$, select a set S of minimum cardinality (called directed dominating set), such that for every vertex $v \in V \setminus S$ there is a directed edge (u, v) where $u \in S$.

Lemma 5.4 *Let* $\texttt{opt}(I), \texttt{opt}(I')$ *be the costs of the optimal solutions of instances* I *and* I' *respectively. Then* $\texttt{opt}(I') \leq \texttt{opt}(I)$*.*

Proof of Lemma 5.4. Let $\texttt{opt}(I) = k$ be the cost of the optimal solution in the MIN-DS$_d$ instance I. This means that there is a set, say S, of k vertices that dominates every other vertex in V. Consider a delegation function which asks every voter $\hat{v}$ such that $v \in S$ to cast a vote. Given that S is a directed dominating set, for every vertex $v \in V \setminus S$, there is at least one directed edge (u, v) with $u \in S$. By the construction of I', $\hat{v}$ approves $\hat{u}$ as her guru and thus, by delegating to $\hat{u}$, she will be satisfied (if there are multiple voters that cast a ballot and are approved by $\hat{v}$, she can delegate arbitrarily to any of them). So, there is a delegation function such that, any voter $\hat{v} : v \in V \setminus S$ is satisfied, resulting in a solution where the number of dissatisfied voters is at most k. Hence, $\texttt{opt}(I') \leq \texttt{opt}(I)$. ⊠

To complete the reduction, we also need to be able to transform a solution of I' to a solution of I.

Lemma 5.5 *For every solution to I' with cost $\mathtt{sol}(I')$, we can find a corresponding solution to I with cost $\mathtt{sol}(I)$, such that $\mathtt{sol}(I) \leq \mathtt{sol}(I')$.*

Proof of Lemma 5.5. Consider a solution to I' of cost $\mathtt{sol}(I')$. Using Proposition 5.1, we can assume that the solution is a collection of disjoint directed stars and using Proposition 5.2 (part (ii)) we can assume that no-one can be dissatisfied because of abstaining or of being represented by a non-approved guru. To summarize, there is a solution of I' of cost $\mathtt{sol}(I')$ in which every voter either casts a vote or delegates to one of her approved gurus and hence the only voters who can be dissatisfied are those who cast a ballot.

Suppose that $\mathtt{sol}(I') = k$, i.e., there is a set S of k dissatisfied voters. For I, we claim that the set of vertices $\{v : \hat{v} \in S\}$ is a directed dominating set. Let v be a vertex that does not belong to that set. Since every $\hat{v} \notin S$ delegates to some voter who casts a ballot, i.e., to some $\hat{u} \in S$, such that $\hat{u} \in N(\hat{v})$, this implies that $(u, v) \in E$. Hence v is dominated by u in G. ⊠

By combining Lemmas 5.4 and 5.5 we have that any α-approximation algorithm for MIN-SC directly implies an α-approximation for MIN-DS$_d$, since we would be able to get a solution to I with: $\mathtt{sol}(I) \leq \mathtt{sol}(I') \leq \alpha \cdot \mathtt{opt}(I') \leq \alpha \cdot \mathtt{opt}(I)$. The proof is now concluded by utilizing the known hardness results for MIN-DS$_d$. In particular, Theorem 10 from [CC08] has established that if G has in-degree bounded by a constant b (which corresponds to Δ for the MIN-SC instance), then MIN-DS$_d$ is $(b-1)$-inapproximable, for $b \geq 3$ and 1.36-inapproximable, for $b = 2$. The logarithmic hardness result for general instances follows from the connection of MIN-DS$_d$ to the classic SET COVER problem. □

Inapproximability results dependent on other parameters can be obtained from the presented reduction as well. For instance, if the in-degree of every voter, say δ, is bounded, then MIN-SC, is $(\ln \delta - \Theta(1) \ln \ln \delta)$-inapproximable, by using a result from [CC08].

Since hardness results have been established for $\Delta \geq 2$, it is natural to question whether an optimal algorithm could be found for the case of $\Delta \leq 1$. This scenario is far from unexciting. Consider for instance a spatial model where voters are represented by points in some Euclidean space, interpreted as opinions on the outcomes of some issues. If each voter approves for delegation only the nearest located voter to her, we have precisely that $\Delta = 1$. The following theorem provides an affirmative answer in the above stated question (its proof is actually a direct Corollary of Theorem 5.14 from Section 5.3).

Theorem 5.14: Consider an instance of ASSL, and let G be its corresponding graph. MIN-SC and MAX-SG are in FPT with respect to the treewidth of G.

Theorem 5.6 *When $\Delta \leq 1$, MIN-SC can be solved in polynomial time.*

Proof. We just need to observe that when $\Delta \leq 1$, the treewidth of the graph of the instance is bounded and thus it is possible to use Theorem 5.14 in order to prove the theorem. Let us describe the structure of the input graph G for the case of $\Delta \leq 1$. For simplicity, suppose that G is (weakly) connected, then

G can be seen as the union of the graphs c and t_i where c is a set of $k \geq 1$ vertices $\{v_1, v_2, \dots, v_k\}$ that, if $k > 1$ form a directed cycle and if $k = 1$ it is just the single vertex v_1 and each t_i for $i = 1, 2, \dots, k$ is either the single vertex v_i or a directed tree rooted at v_i having all the edges towards the root. Hence, the treewidth of the input graph is no more than 2, which means that the Algorithm of Theorem 5.14 results to an optimal solution in polynomial time for the problem of minimizing the number of dissatisfied voters. □

For higher values of Δ, we can only hope for approximation algorithms. As we show next, it is possible to complement Theorem 5.3 with asymptotically tight approximation guarantees by reducing MIN-SC to the SET COVER problem.

Theorem 5.7 *Let* $\Delta = \max_{v \in V} |N(v)| \geq 2$. *When* Δ *is constant there is a polynomial time algorithm for MIN-SC with a constant approximation ratio of* $(\Delta + 1)$. *For general instances, the problem is* $(\ln n - \ln \ln n + \Theta(1))$*-approximable.*

Proof. We will present a reduction that preserves approximability to the SET COVER problem. From an instance I of MIN-SC we create an instance I' of SET COVER as follows: We create a universe of elements U by adding one element for every voter, except for certain voters for which there is no such need. In particular, U contains one element for every $v \in V \setminus (C \cup A \cup \{u : \exists u' \in N(u) \cap C\})$. This means that in U we have excluded voters who can be satisfied without delegating to someone else as well as voters who can be satisfied by delegating to members of C (observe that because of Proposition 5.2 (part (*i*)), all voters of C will be assigned to vote). Furthermore, to describe the collection $\mathcal{F}$ of sets in I', for every voter $v \in V \setminus C$ we add the set $S_v = U \cap (\{u : v \in N(u)\} \cup \{v\})$. If some S_v turns out to be the empty set, it can be simply disregarded (e.g. for a voter v with $N(v) \cap C \neq \emptyset$).

In an unweighted SET COVER instance, we are given a universe U and a collection $\mathcal{F}$ of subsets of U, and ask to find a cover of the universe with the minimum possible number of sets from $\mathcal{F}$.

Proposition 5.2: Consider a solution given by a delegation function $d(\cdot)$. There always exists a solution which is at least as good (i.e., the number of satisfied voters is at least as high) as $d(\cdot)$, in which (i) every voter in C casts a ballot and (ii) if any voter is dissatisfied, it is because she is casting a ballot without approving it.

Lemma 5.8 *Let* $\texttt{opt}(I), \texttt{opt}(I')$ *be the costs of the optimal solutions in the instances I and* I' *respectively. Then* $\texttt{opt}(I') \leq \texttt{opt}(I)$.

Proof of Lemma 5.8. Let there be k dissatisfied voters in the optimal solution of the MIN-SC instance I. By making use of Proposition 5.2, we can assume that these are members of $V \setminus C$ who are assigned to cast a ballot. Hence, for a dissatisfied voter v there exists a corresponding set S_v in I'. We will argue that by selecting these k sets that correspond to dissatisfied voters, we have a feasible solution for the SET COVER problem. Towards contradiction, assume that there is an element in I' that has not been covered by any of these sets. Because of the definition of U, there must exist a voter v in I who corresponds to that element and who only accepts to delegate to some voters who are not in C, i.e., each $u \in N(v)$ has a corresponding set S_u in $\mathcal{F}$ since

$u \in V \setminus C$. Moreover, v should be satisfied in I, otherwise the set S_v would have been selected in the solution we constructed for I' and v would have been covered. Therefore, at least one of her approved voters, say u, is a guru, and the set S_u covers v, which is a contradiction. ⊠

Lemma 5.9 *Given a feasible solution with cost* $\mathtt{sol}(I')$ *of the produced instance* I'*, we can create a feasible solution of* I*, with cost* $\mathtt{sol}(I) \leq \mathtt{sol}(I')$.

Proof of Lemma 5.9. Say that $\mathtt{sol}(I') = k$, which means that by selecting a number of k sets, it is possible to cover every element of U. Consider a delegation function $d(\cdot)$ which asks every voter from $V \setminus C$ whose corresponding set has been selected in the cover, to cast a ballot. Following Proposition 5.2, it also asks every voter from C to cast a ballot. From these, only the former k voters are dissatisfied, who vote but do not belong to C. We will argue that we can make all the remaining voters satisfied and hence we will have a solution with k dissatisfied voters.

Proposition 5.2: Consider a solution given by a delegation function $d(\cdot)$. There always exists a solution which is at least as good (i.e., the number of satisfied voters is at least as high) as $d(\cdot)$, in which (i) every voter in C casts a ballot and (ii) if any voter is dissatisfied, it is because she is casting a ballot without approving it.

Consider a voter $v \in V \setminus C$, whose set S_v was not included in the SET COVER solution. If $v \in A$, then v is assigned to abstain and she is satisfied. So, suppose that $v \in V \setminus (A \cup C)$ and also that $N(v) \neq \emptyset$ (otherwise, with $N(v) = \emptyset$, then S_v would have been selected in the cover).

Case 1: $N(v) \cap C \neq \emptyset$. Then v can delegate to a member of C and be satisfied.

Case 2: $N(v) \cap C = \emptyset$. Then by the construction of the universe U, we have that $v \in U$. Since we have selected a cover for U, v is covered by some set. Additionally, we have assumed that S_v was not picked in the cover, hence v is covered by some other set, say S_u, which means that u is a voter who is assigned to cast a vote and $v \in S_u$. But then v can delegate to u and be satisfied.

Hence, we have identified a solution with k voters being dissatisfied. ⊠

By combining Lemma 5.8 and Lemma 5.9, we have that if we run any α-approximation algorithm for the SET COVER instance I', we can find a solution for the MIN-SC instance I, with the same guarantee since $\mathtt{sol}(I) \leq \mathtt{sol}(I') \leq \alpha \cdot \mathtt{opt}(I') \leq \alpha \cdot \mathtt{opt}(I)$. Recall that there exists a well known f-approximation algorithm for SET COVER, where f is the maximum number of sets that contain any element. Note also that in our construction, each element of I' that corresponds to a voter v of I, belongs to at most $|N(v)| + 1$ sets. This directly yields a $(\Delta + 1)$-approximation for our problem. Alternatively, when Δ is not bounded, we can use the best currently known approximation algorithm for the SET COVER problem [Sla97], to obtain the desired result. □

[Sla97] Slavık (1997): A Tight Analysis of the Greedy Algorithm for Set Cover.

5.2.2 Social Good Maximization

In all voting problems that involve a notion of dissatisfaction, one can study either minimization of dissatisfactions or maximization of satisfactions. The minimization version is slightly more popular, see e.g., [EGP19] (also, in approval voting elections, it is more common to minimize the sum of distances from the optimal solution than to maximize the satisfaction score). Clearly, for ASSL, if we can solve optimally MIN-SC, the same holds for MAX-SG. The problems however can differ on their approximability properties.

Looking back on our findings for MIN-SC, we note that the results from Theorem 5.3 immediately yield NP-hardness for MAX-SG. The hardness of approximation however does not transfer. The result of Theorem 5.6 also applies.

Theorem 5.3: Let $\Delta = \max_{v \in V} |N(v)| \geq 2$. When Δ is constant, it is NP-hard to approximate MIN-SC with a ratio smaller than $\max\{1.36, \Delta - 1\}$. For general instances, it is NP-hard to achieve an approximation better than $\ln n - \Theta(1) \ln \ln n$.

Theorem 5.6: When $\Delta \leq 1$, MIN-SC can be solved in polynomial time.

Corollary 5.10 *Let $\Delta = \max_{v \in V} |N(v)|$. Then MAX-SG is NP-hard even when $\Delta = 2$, and it is efficiently solvable when $\Delta \leq 1$.*

Next, we also provide a constant factor approximation for constant Δ, albeit with a worse constant than the results for MIN-SC. The main insight for the next theorem is that we can exploit results from the rich domain of Constraint Satisfaction Problems (CSPS) and model MAX-SG as such.

Theorem 5.11 *Let $\Delta = \max_{v \in V} |N(v)| \geq 2$. When $\Delta \in O(1)$ there is a polynomial time algorithm for MAX-SG with an approximation ratio of $\frac{1}{(\Delta+2)^{\Delta+2}}$.*

Proof. We will use an approximation preserving reduction of an instance I of MAX-SG to an instance I' of MAX k-CSP(d) and we will use an approximation algorithm for the latter to prove the statement.

MAX k-CSP(d): Given a set of constraints each of which depends on at most k variables and such that each variable can have its own domain, which is of size at most d, find an assignment to the variables that maximizes the number of satisfied constraints.

For every voter v of I, we create a variable x_v of I' having as possible values the set $dom(x_v) = N(v) \cup \{v, \perp\}$. Hence, if $|N(v)| \leq \Delta$ for every voter v, the domain size in I' is at most $\Delta + 2$ for every variable. We add in I' one constraint C_v for each voter v, specifying the cases where v is satisfied. The constraint C_v can be viewed as the logical OR of the following terms:

- If $v \in C$, we add in C_v the term $(x_v = v)$.
- If $v \in A$, we add in C_v the term $(x_v = \perp)$.
- For any voter $u \in N(v)$, we add the term $(x_v = u) \wedge (x_u = u)$.

In order to provide each constraint as part of the input of I', it suffices to provide the truth table or all the combinations of assignments to the relevant variables that make the constraint satisfied. Since we have assumed that $\Delta \in O(1)$, no more than a constant number of $O(\Delta^{\Delta})$ combinations are needed.

Any solution of I' is an assignment to every variable x_v of a value from $dom(x_v)$, which can be directly translated to a solution of I. We observe that the constraint C_v is satisfied if at least one of the terms we added in C_v is satisfied, which is true if and only if voter v is satisfied. Thus, if an assignment in the variables of I' satisfies q constraints, there are exactly q satisfied voters in the instance I. Hence by running an α-approximation algorithm for MAX k-CSP(d) we can obtain a solution that satisfies an α-fraction of the maximum possible number of satisfied voters.

The constraint that corresponds to voter v has at most $\Delta + 2$ clauses and the domain of each variable is $\Delta + 2$. An instance of MAX k-CSP(d), can be approximated within a factor of $\frac{1}{d^k}$ [CCJ+05] (Proposition 2.3 therein). □

We leave as an open problem the question of whether there exist better approximations or whether one can establish hardness of approximation results.

In case we are willing to settle with a randomized algorithm, we can have a better approximation with an expected ratio of $\Omega(\frac{\max\{k, logd\}}{d^{k-1}})$ [MM17], using the results of [MM12] and [MNT16].

5.2.3 Further Implications: Instances with Bounded Social Cost

We conclude this section by discussing some implications that can be derived by the reductions presented in Section 5.2.1, on relevant questions to MIN-SC. Let us start with the special case where the optimal cost of an instance I is zero, i.e., it is possible to satisfy all voters. Can we have an algorithm that detects this? It would be ideal to compute a delegation function that does not cause any dissatisfactions, and this is indeed possible. If $\mathtt{opt}(I) = 0$, then any approximation algorithm for MIN-SC of finite ratio will necessarily return an optimal solution. If $\mathtt{opt}(I) > 0$, the approximation algorithm will also return a positive cost. Hence, by using Theorem 5.7 we have the following:

Theorem 5.7: Let $\Delta = \max_{v \in V} |N(v)| \geq 2$. When Δ is constant there is a polynomial time algorithm for MIN-SC with a constant approximation ratio of $(\Delta + 1)$. For general instances, the problem is $(\ln n - \ln \ln n + \Theta(1))$-approximable.

Corollary 5.12 *Given an instance of ASSL, there exists a polynomial time algorithm that decides if it is possible to satisfy all voters, in which case it can also construct an optimal delegation function.*

Taking it a step further, suppose now we ask: Given an instance I, is it true that $\mathtt{opt}(I) \leq k$, for some positive constant k? This time, we can construct a SET COVER instance I' using the reduction presented in the proof of Theorem 5.7, and then we can enumerate all possible collections of subsets of size at most k. If a solution is found, it corresponds to a set of at most k dissatisfied voters. Hence, we can solve the problem in time $n^{O(k)}$. But now we can question whether there is hope for a substantially better running time. To answer this, we exploit the reduction used in Theorem 5.3 from DIRECTED DOMINATING SET. In particular, it is known by [GV08] that DOMINATING SET is $W[2]$-hard when parameterized by the solution cost, even in graphs of bounded average degree. Given that the directed version of DOMINATING SET inherits the hardness results of the undirected version in combination with the proof of Theorem 5.3, we have the following.

[GV08] Golovach and Villanger (2008): Parameterized Complexity for Domination Problems on Degenerate Graphs.

Corollary 5.13 *MIN-SC cannot be solved in time $f(k)n^{O(1)}$ even for the case where Δ is constant, where $f(k)$ is a computable function depending only on the minimum possible number k of dissatisfied voters, unless $W[2] = FPT$.*

5.3 Exact Algorithms via Monadic Second Order Logic

The goal of this section is to focus on special cases that admit exact polynomial time algorithms. Our major highlight is the use of a logic-based technique for obtaining such algorithms. To our knowledge, this framework has not received much attention (if at all) from the Computational Social Choice community despite its wide applicability on graph-theoretic problems. We therefore expect that this has the potential of further deployments for other related problems.

5.3.1 Optimization under Bounded Treewidth

The general methodology involves the use of an algorithmic meta-theorem (for related surveys see [Gro08] and [Kre08]) to check the satisfiability of a formula that expresses a graph property, defined over an input graph of bounded treewidth. Roughly speaking, the treewidth is a graph parameter that indicates the "tree-likeness" of a graph. It was introduced independently by various authors mainly for undirected graphs (see [Sey14] for an extended exposition of the origin of the notion) but its definition and intuition can be extended to directed graphs as well [ALS91]. In our case, we will require bounded treewidth for the directed graph associated to an instance of ASSL.

[Gro08] Grohe (2008): Logic, Graphs, and Algorithms.

[Kre08] Kreutzer (2008): Algorithmic Meta-Theorems.

[Sey14] Seymour (2014): The Origin of the Notion of Treewidth.

[ALS91] Arnborg et al. (1991): Easy Problems for Tree-Decomposable Graphs.

A Brief Introduction to Tractability Results via MSO Logic

The approach discussed here was initiated by Courcelle [Cou90], who used Monadic Second Order (MSO) logic to define graph properties. These, typically ask for some set of vertices or edges subject to certain constraints. For expressing a property in MSO, we can make use of variables for edges, vertices as well as for subsets of them. Apart from the variables[1] we can also have the usual boolean connectives $\neg, \wedge, \vee, \Rightarrow$, quantifiers $\forall, \exists$, and the membership operator $\in$. The resulting running time for deciding properties expressible in MSO turns out to be exponentially dependent on the treewidth and the size of the formula.

After Courcelle's theorem, there have been several works that extend the algorithmic implications of MSO logic. Most importantly, and most relevant to us, the framework of [ALS91] can handle some types of optimization problems. Consider a formula $\phi(X_1, \dots, X_r)$ in MSO, having $X_1, \dots, X_r$ as free

[Cou90] Courcelle (1990): The Monadic Second-Order Logic of Graphs. I. Recognizable Sets of Finite Graphs.

[1] For ease of presentation, we will also use set operations that although they are not explicitly allowed, they can be easily replaced by MSO expressions (e.g., $x \notin A \setminus B \equiv \neg((x \in A) \wedge \neg(x \in B))$ and $A \subseteq B \equiv (\forall x \in A \Rightarrow x \in B)$).

set variables, so that a property is true if there exists an assignment to the free variables that make ϕ satisfied. Then, we can optimize a weighted sum over elements that belong to any such set variable, subject to the formula ϕ being true (one needs to be careful though as the weights are taken in unary form). A representative example presented in [ALS91] (see Theorem 3.6 therein for a wide variety of tractable problems with respect to treewidth) is MINIMUM DOMINATING SET in which we want to minimize $|X|$ subject to a formula that enforces the set X to be a dominating set.

We note that the results we use here require to have a representation of the tree decomposition of the input graph. But even if this is not readily available, its computation is in FPT with respect to the treewidth [Bod96].

[Bod96] Bodlaender (1996): A Linear-Time Algorithm for Finding Tree-Decompositions of Small Treewidth.

Our first result in this section shows that MIN-SC and MAX-SG are tractable when the treewidth of the associated graph is constant.

Theorem 5.14 *Consider an instance of ASSL, and let G be its corresponding graph. MIN-SC and MAX-SG are in FPT with respect to the treewidth of G.*

Proof. It suffices to solve MIN-SC since this yields an optimal solution to MAX-SG as well. In order to apply a framework of MSO logic, we first make a small modification to the graph G. We add a special vertex denoted by a and we add a directed edge (v, a) for every v for which $v \in A$. In this manner, abstainers will be encoded by "delegating" their vote to a. Let $G' = (V', E')$ be the resulting graph, where $V' = V \cup \{a\}$ and $E' = E \cup \{(v, a) : v \in A\}$. We observe that these additions do not affect the boundedness of the treewidth.

Lemma 5.15 *If G has bounded treewidth, so does G'.*

We will create an MSO formula $\phi(D, X)$ with 2 free variables, D and X, encoding an edge-set and a vertex set respectively. The rationale is that $\phi(D, X)$ becomes true when the edges of D encode a delegation function and X denotes the set of voters who are dissatisfied by the delegations of D. To write the formula, we also exploit the fact that the framework of [ALS91] allows the use of a constant number of "distinguished" sets so that we can quantify over them as well (apart from quantification over V' and E'). We will use V, along with C and A (of voters who approve casting a ballot or abstaining respectively), as these special sets. To proceed, $\phi(D, X)$ is the following:

[ALS91] Arnborg et al. (1991): Easy Problems for Tree-Decomposable Graphs.

$$
\begin{aligned}
&D \subseteq E' \wedge X \subseteq V \setminus C \wedge \\
&(\forall v \in V' \ (deg^+_D(v) \leq 1)) \wedge \\
&(\forall u, v, w \in V'((u, v) \in D \Rightarrow (v, w) \notin D)) \wedge \\
&(\forall v \in C \ (deg^+_D(v) = 0)) \wedge \\
&(\forall v \in V \ (v \in X \Leftrightarrow (deg^+_D(v) = 0 \wedge v \notin C)))
\end{aligned}
$$

The term $deg^+_D(v) = 0$ can be expressed in MSO logic in a similar way to the more general term of $deg^+_D(v) \leq 1$, which we define formally as

$$
\begin{gathered}
deg^+_D(v) \leq 1 \equiv \\
(\exists u \in V' \ (v, u) \in D) \Rightarrow \neg(\exists w \in V' \ ((v, w) \in D \wedge w \neq u)).
\end{gathered}
$$

Concerning the construction of $\phi(D, X)$, the second line expresses the fact that D is a union of disjoint directed stars so as to enforce Proposition 5.1. Anyone with out-degree equal to one within D either delegates to some other voter or abstains (i.e. delegates to vertex a), whereas those with out-degree equal to zero in D cast a vote themselves. The third line of $\phi(D, X)$ also enforces Proposition 5.2 (part (i)) so that members of C always cast a vote. The fourth line expresses the fact that the vertices of X are dissatisfied voters. By Proposition 5.2 (part (ii)), the only way to make a voter v dissatisfied is by asking her to cast a ballot when $v \notin C$. Indeed, voters who are not asked to cast a ballot, have out-degree equal to one in D, so they either abstain or delegate. This means that either $(v, a) \in D$ or $(v, u) \in D$ for some $u \in V$. In the former case, v is satisfied because $v \in A$ (if $v \notin A$ then the edge (v, a) would not exist in E' and could not have been selected in D). In the latter case, v approves u (otherwise the edge (v, u) would not exist) and u casts a vote since D contains only stars. Hence v is again satisfied.

The final step is to perform optimization with respect to $|X|$ subject to $\phi(D, X)$ being true. To that end, we can assign a weight $w(v)$ to every vertex v such that $w(a) = 0$ and $w(v) = 1, \forall v \in V' \setminus \{a\}$. Hence $\sum_{v \in X} w(v) = |X|$. Using the result of [ALS91], we can find a delegation function $d(\cdot)$, as given by the edges in D, that minimizes the number of dissatisfied voters within the feasible solutions. □

Proposition 5.1: Consider a solution given by a delegation function $d(\cdot)$. There always exists a solution $d'(\cdot)$ which is at least as good (i.e., the number of satisfied voters is at least as high) as $d(\cdot)$, so that $G(d')$ is a collection of disjoint directed (towards the central vertex) stars, and voters that abstain form isolated vertices.

Proposition 5.2: Consider a solution given by a delegation function $d(\cdot)$. There always exists a solution which is at least as good (i.e., the number of satisfied voters is at least as high) as $d(\cdot)$, in which (i) every voter in C casts a ballot and (ii) if any voter is dissatisfied, it is because she is casting a ballot without approving it.

5.3.2 Adding Secondary Objectives

We continue with exhibiting that MSO frameworks can be useful for tackling other related problems as well. For the cases when we can solve MIN-SC (and MAX-SG) optimally, we are investigating whether we can find such a solution with additional properties (whenever the optimal is not unique). Motivated by questions studied in [EGP19; EGP20; GKM+21] we consider the set of problems presented below:

[EGP19] Escoffier et al. (2019): The Convergence of Iterative Delegations in Liquid Democracy in a Social Network.

[EGP20] Escoffier et al. (2020): Iterative Delegations in Liquid Democracy with Restricted Preferences.

[GKM+21] Gölz et al. (2021): The Fluid Mechanics of Liquid Democracy.

1. Among the optimal solutions to MIN-SC (or MAX-SG), find one in which a given voter v casts a vote, or answer that no such solution exists.
2. Ditto, with having voter v abstain in an optimal solution.
3. Among the optimal solutions to MIN-SC (or MAX-SG), find one that minimizes the number of abstainers.
4. Among the optimal solutions to MIN-SC (or MAX-SG), find one that minimizes the maximum voting power over all gurus, i.e. the number of voters that she represents (or equivalently that minimizes the maximum in-degree).

The fourth problem is quite important in models of LD, given also the critique that often applies on such models that may accumulate excessive power on some voters. Below we start by addressing the first three problems together.

Theorem 5.16 *Consider an instance of ASSL, and let G be its corresponding graph. It is in FPT with respect to the treewidth of G to find an optimal solution to MIN-SC and MAX-SG, in which a given voter casts a ballot or abstains (if such a solution exists). The same holds for minimizing the number of abstainers.*

Proof. It suffices to solve MIN-SC since this yields an optimal solution to MAX-SG as well. First of all, we note that it is safe to focus only on solutions that are formed by disjoint stars since we can argue, similarly to Proposition 5.1, that such a restriction will not affect the optimal solutions we are looking for in any of these problems. Let $G' = (V', E')$ be the graph used in the proof of Theorem 5.14 and let $\phi(D, X)$ be the MSO formula produced in that proof.

For the first problem, let s be the given voter so that we want to check if there exists an optimal solution where s casts a ballot. A small modification is only required for the case that s is needed to abstain, the proof of which we are going to omit.

We begin by using Theorem 5.14 to find the number of dissatisfied voters in an optimal solution to MIN-SC. Let c be this number. If s is among those who vote themselves in the solution we found, we are done. If not, our goal is to see if we can find another optimal solution where s is assigned to vote. To this end, we define a new formula, $\phi_1(D, X)$ as follows:

$$\phi_1(D, X) \equiv \phi(D, X) \wedge (deg^+_D(s) = 0).$$

The formula $\phi_1(D, X)$ becomes true when D encodes a delegation function under which s casts a ballot and X is the set of dissatisfied voters.

We now employ the use of an *evaluation relation* from the framework of [ALS91]. An evaluation relation for a free variable Y is a boolean condition that concerns the weighted sum of elements belonging to Y. Therefore, instead of optimizing as we did in the proof of Theorem 5.14, we can ask if an evaluation relation holds subject to the formula ϕ_1. In particular, we use the weights that we used in Theorem 5.14, and add the evaluation relation $\sum_{v\in X} w(v) \leq c$, which is equivalent to $|X| \leq c$. The algorithm of [ALS91] for this formulation, will either return a delegation function with c dissatisfied voters, and where s casts a ballot, or it will answer negatively, meaning that any solution in which s votes, dissatisfies more than c voters and thus could not be optimal.

For the problem of finding an optimal solution with the minimum possible number of abstainers, as before, we will first solve MIN-SC using Theorem 5.14 and let c be the cost of the optimal solution.

We now define the following formula $\phi_2(D, X, Y)$, where we have added a free variable Y to denote the set voters that are forced to abstain.

$$\phi_2(D, X, Y) \equiv \phi(D, X) \wedge (Y \subseteq V') \wedge (v \in Y \Leftrightarrow (v, a) \in D).$$

Finally, we add the evaluation relation $(|X| \leq c) \wedge (|Y| \leq b)$ and run the algorithm of [ALS91] for every possible value of $b \in \{0, 1, \dots, n\}$, starting from $b = 0$. The first value of b for which we will have a satisfying assignment for the evaluation relation subject to ϕ_2 being true, yields a delegation function with the minimum possible number of abstainers. □

We come now to the fourth problem, which is the most challenging one. For this, we will use yet another enriched version of the MSO framework, which facilitates the addition of further constraints and helps in solving several degree-constrained optimization problems. As these problems are in general more difficult [MT20], the results of [Sze11] and [KKM+19] yield polynomial time algorithms with respect to treewidth, but do not place them in FPT.

For the presentation we will stick to the terminology of [KKM+19]. Consider a formula $\phi(X_1, \dots, X_r)$ with free variables $X_1, \dots X_r$. The main idea is to add so-called global and local cardinality constraints and ask for an assignment that satisfies both ϕ and the constraints. In the simpler version that we will use here, a global cardinality constraint is of the form $\sum_{i\in[r]} a_i|X_i| \leq b$ for given rational numbers $a_i, i \in [k]$ and b (some of these numbers can be zero so that we constrain the cardinality of only some of the free variables). On the other hand, a local cardinality constraint for a vertex has to do with limiting the number of its neighbors or incident edges that belong to a set corresponding to a free variable. For example, if X_1 is a free variable of ϕ that encodes a vertex set, and X_2 is a free variable encoding an edge set, we can have constraints of the form "for each vertex v of G, the number of vertices in X_1 adjacent to v belongs to a set $a(v)$", where $a(v)$ contains the allowed values (e.g., could be an interval). Similarly, we can express that the number of edges of X_2 incident with v can take

[Sze11] Szeider (2011): Monadic Second Order Logic on Graphs with Local Cardinality Constraints.

[KKM+19] Knop et al. (2019): Simplified Algorithmic Metatheorems Beyond MSO: Treewidth and Neighborhood Diversity.

only specific values from some set $a'(v)$. A representative illustration for local constraints in [KKM+19] is the CAPACITATED DOMINATING SET problem, where one needs to pick a dominating set respecting capacity constraints.

Theorem 5.17 *Consider an instance of* ASSL *where the associated graph G has constant treewidth. Among the optimal solutions to* MIN-SC*, we can find in polynomial time a solution minimizing the maximum in-degree of the gurus.*

Proof. First of all, similarly to the proof of Theorem 5.16, it suffices to focus only on solutions that are formed by disjoint directed stars. Let $G' = (V', E')$ be the graph used in the proof of Theorem 5.14, derived from G. Our first step is to use Theorem 5.14 and solve MIN-SC optimally so that we know the cost of an optimal solution. Suppose that we have c unsatisfied voters in an optimal solution.

To proceed, we will utilize the extended MSO framework of [KKM+19], but to do so, we need to work with an undirected graph.[2] To this end, we will create an undirected graph $H = (V'', E'')$ from the directed graph G' as follows:

- The set V'' will contain 2 sets of vertices, V_{in}, V_{out}, so that for every vertex $v \in V'$ such that $v \neq a$ we create a vertex $v_{in} \in V_{in}$ and a vertex $v_{out} \in V_{out}$. The vertex v_{in} will be used to indicate connections with in-neighbors in G', whereas v_{out} will be used to indicate connections with out-neighbors in G'. We also add the vertex a in V'' with the same meaning as in G', so that in total $V'' = V_{in} \cup V_{out} \cup \{a\}$.
- The set E'' will contain 2 sets of edges, E_d, E_p, so that in the set E_d, for every directed edge $(u, v) \in E'$, we add an undirected edge (u_{out}, v_{in}), if $v \neq a$, or we add an edge (u_{out}, a) in it otherwise. We also create a set E_p which for every $v \in V'$ includes an undirected edge between the pair (v_{in}, v_{out}). We note that the set E_p is added so that we can reference a vertex v_{out} when given v_{in}, in the MSO formula that we will produce. Thus, in total: $E'' = E_d \cup E_p$.

Hence, the graph H can be described by the sets $V_{in}, V_{out}, E_d, E_p$. We will make use of some of these distinguished sets in the MSO formula. Given the construction, we can verify that the treewidth of H is no more than a constant multiple of the treewidth of G. That is because if we consider a tree-decomposition of G and we replace each vertex v in it with two vertices v_{in} and v_{out}, we end up having a valid tree-decomposition of H with at most twice the treewidth of G. Hence the following holds:

Lemma 5.18 *If G has bounded treewidth, so does H.*

[2] It is not obvious if the results of [KKM+19] can be applied right away to directed graphs.

For notational convenience, we define the predicate $out(v, u)$ to be true exactly when $(u, v) \in E_p$, i.e., v and u are the in-vertex and out-vertex respectively of some v' from G. We will now produce a formula for the undirected graph H, whose satisfying assignments will correspond to valid delegation functions on the original graph G. We denote our formula by $\psi(D, F, X)$, where the free variables D, F, X have the following interpretation:

- As in Theorem 5.14, the set D will be a subset of edges from H. By the definition of H, D will contain either edges between V_{in} and V_{out}, e.g., of the form (v_{out}, u_{in}) for some $v, u \in V$, so that they encode a delegation from v to u in G or they will have edges from V_{out} to the vertex a encoding abstention.
- The set F will encode the set of voters who cast a ballot themselves. We make the convention that F contains only vertices from V_{in}, i.e., to encode that a voter $v \in V$ votes, it should hold that $v_{in} \in F$ and that the degree of v_{out} in D must be 0.
- The set X will encode the dissatisfied voters induced by D. Again, we make the convention that X will be a subset of V_{in}.

Given the above interpretation, the formula $\psi(D, F, X)$ will be:

$$
\begin{aligned}
&(D \subseteq E'' \setminus E_p) \wedge (X \subseteq V_{in}) \wedge (C \subseteq F \subseteq V_{in}) \wedge \\
&\forall v \in F\ \exists u \in V_{out}\ (out(v, u) \wedge deg_D(u) = 0) \wedge \\
&\forall v \in V_{in} \setminus F\ \exists u \in V_{out} : \\
&\big[out(v, u) \wedge deg_D(u) = 1 \wedge deg_D(v) = 0 \wedge \\
&((u, a) \in D \vee \exists u' \in F\ (u, u') \in D)\big] \wedge \\
&\forall v \in V_{in}\ (v \in X \Leftrightarrow (v \notin C \wedge v \in F))
\end{aligned}
$$

For all terms in the formula above that involve the degree of a vertex, e.g. $deg_D(u) = 0$, we refer to the proof of Theorem 5.14 for writing them formally in MSO logic.

Following the framework of [KKM+19], we now add 2 classes of constraints that we want to be satisfied in addition to $\psi(D, F, X)$. The first one is a so-called global cardinality constraint to ensure that the number of dissatisfied voters is optimal. Here, a global cardinality constraint will be in the form $a_1|D| + a_2|F| + a_3|X| \leq r$, for a given integer r. Since we have already solved MIN-SC and the solution is c, and since X expresses the set of dissatisfied voters, we choose $a_1 = a_2 = 0$, $a_3 = 1$ and demand that $|X| \leq c$.

[KKM+19] Knop et al. (2019): Simplified Algorithmic Metatheorems Beyond MSO: Treewidth and Neighborhood Diversity.

Finally, we add the so-called local cardinality constraints. We will produce a set of constraints depended on a fixed number $d \in \{1, \dots, n\}$ such that the constraints will ensure that the maximum degree of every vertex in F is bounded by d or equivalently that no voter receives more than d delegations (apart from the special vertex a, the degree of which is not needed to be bounded). To do this, let $I(v)$ denote the set of edges incident with a vertex v in H. Then the constraints can be written in the form:

$$\forall v \in V : |D \cap I(v)| \in [0, B(v)], \text{where } B(v) = \begin{cases} n, \text{if } v = a, \\ d, \text{if } v \in V_{in}, \\ 1, \text{if } v \in V_{out}. \end{cases}$$

By using [KKM+19], we can now decide for every d if there is an assignment to the variables D, F, X that satisfies $\psi(D, F, X)$ together with the global and local cardinality constraints. For the first value of d where this is true, we can directly translate the solution to a delegation function. In particular, every edge in D corresponds immediately to a delegation or abstention in our original instance, whereas the set F describes the voters who cast a ballot.

To summarize, the suggested algorithm is as follows:

Step 1. Use Theorem 5.14 to find an optimal solution, say with cost c.

Step 2. Transform G to the undirected graph H, construct $\psi(D, F, X)$.

Step 3. For $d = 1$ to n, decide if the formula $\psi(D, F, X)$ is satisfiable subject to the global and local cardinality constraints introduced above. Stop in the first iteration where this is true and create the delegation function from the sets D and F. □

5.4 Concluding Discussion and Future Directions

In this chapter we presented a model that allows voters to express preferences over delegations via an approval set. Our main goal has been to optimize the overall satisfaction of the voters, which implies that it suffices to focus only on direct delegations to actual voters. Even under this simpler solution space, the problems we study are intractable, even when the out-degree is a small constant. On the positive side, we have exhibited constant factor approximation algorithms for graphs of constant maximum out-degree, as well as exact algorithms under the bounded treewidth condition, even when secondary objectives are also present. It is therefore interesting to see if any other parameter can play a crucial role on the problem's computational complexity.

All our results also hold under the generalized model where a graph G is given so that the out-neighborhood $N(v)$, of voter v, expresses the set of feasible delegations which is a (possibly strict) superset of her approved delegations. On the other hand, the case where the approved delegatees of a voter v are not necessarily neighbors seems more complex (e.g., a voter approves some other person but cannot directly delegate to her due to hierarchy constraints). Finally, the results of Section 5.2 also hold for weighted voters, whereas the results of Section 5.3 only hold if the weights are polynomially bounded in unary form.

Another worthwhile direction comes from the fact that the MSO framework primarily serves as a theoretical tool for placing a problem in a certain complexity class but yields impractical running times. One could proceed with a theoretical and/or experimental study of tailor-made dynamic programming algorithms for the problems presented in Section 5.3. Coming to our last result (Theorem 5.17), an interesting approach for future work would be to provide algorithms with trade-offs between the total dissatisfaction and the maximum voting power (instead of optimizing one objective and keeping the other as a secondary).

5.4.1 Towards a Game-Theoretic Analysis

We conclude with a preliminary game-theoretic analysis, which can serve as the basis for a more elaborate future study of these models. Motivated by the approach of [EGP19; EGP20], we define the following simple game: Say that in an instance of ASSL each voter v acts as a strategic player, whose strategy space is $N(v) \cup \{v, \bot\}$. The utility that she can earn from an outcome is either 1, if she is satisfied with that outcome, or 0 otherwise. The first relevant question is whether such games admit pure Nash equilibria, i.e., delegation functions under which no voter is able to unilaterally change her strategy and increase her utility. In contrast to the model of rank-based preferences of [EGP19; EGP20], in our case, Nash equilibria are guaranteed to exist.

[EGP19] Escoffier et al. (2019): The Convergence of Iterative Delegations in Liquid Democracy in a Social Network.

[EGP20] Escoffier et al. (2020): Iterative Delegations in Liquid Democracy with Restricted Preferences.

Proposition 5.19 *In every instance of ASSL, there exists a pure Nash equilibrium, which can be computed in polynomial time.*

Proof. Let I be an arbitrary instance and let $d(\cdot)$ be the following delegation function which we will prove that produces a Nash Equilibrium.

$$d(v) = \begin{cases} v, \text{if } v \in C, \\ u, \text{if } \exists u \in N(v) \cap C, \\ \bot, \text{otherwise.} \end{cases}$$

For the second case above, if there are multiple such nodes u, we pick one arbitrarily. Under d, every $v \in C$ is satisfied, and the same for every voter who approves as her guru a member from C. Thus such voters do not have any incentive to deviate. For the remaining voters, there are some satisfied (those who accept abstaining) and some dissatisfied (those who do not accept abstaining). Consider a voter v who is dissatisfied, and hence $v \notin A$. Suppose that v can change her strategy and become satisfied. Since v was dissatisfied before, $d(v) = \bot$. If she decides to cast a ballot in order to improve her utility, she will still remain dissatisfied since $v \notin C$ (if v belonged to C, then $d(v) = v$ and she would have been satisfied from the beginning).

Similarly, if she decides to delegate to a guru voter u, it must hold that $u \in C$ and at the same time it must hold that v accepts u as her guru. But in that case $d(v)$ would have been equal to u by the definition of $d(\cdot)$, and she would have been satisfied, a contradiction. Hence, no voter has an incentive to unilaterally deviate, and $d(\cdot)$ produces a pure Nash equilibrium. □

In order to evaluate the equilibria of a game (in terms of the derived social good, or similarly in terms of social cost), we can use the Price of Anarchy as a standard metric. This can be defined as the worst possible ratio between the optimal solution for the social good against the number of satisfied voters at a Nash Equilibrium. Unfortunately, we show below that strategic behavior can lead to quite undesirable solutions and we note that this could act as an argument in favor of using a centralized mechanism, as done in the previous sections, to avoid such bad outcomes.

Proposition 5.20 *The Price of Anarchy for the strategic games of the* ASSL *model, can be as bad as $\Omega(n)$, even when $\Delta \leq 1$.*

Proof. We need to prove that there is an instance having n voters as strategic players, in which the number of satisfied voters given by an equilibrium is $\Omega(n)$ times worse than the number of satisfied voters given by the optimal solution. For this, consider an instance I, with n voters, one of which, namely voter c only accepts abstaining and all the rest only accept delegating to c. Note that the graph of the instance is a directed star having edges towards c.

It is clear that the optimal solution is to have c vote and have all other voters delegate to c. In this way we have that $\texttt{opt}(I) = n - 1$, since only c is dissatisfied. Let $d(\cdot)$ now be the delegation function such that $d(v) = c$, if $v \neq c$ and $d(c) = \perp$.

We claim that d induces an equilibrium. Indeed, since c is abstaining, no other voter can affect the outcome by deviating to some other strategy. Hence, we have found an equilibrium with only one satisfied voter. □

Finally, Proposition 5.20 raises the question of coming up with richer game-theoretic models of the delegation process (e.g. richer utility functions or repeated games) so as to understand thoroughly the effects of strategic behavior in the proposed framework of LD.

As Time Goes By: Adding a Temporal Dimension to Resolve Delegations in Liquid Democracy 6

As discussed in Chapter 5, Liquid Democracy (LD) is a voting framework that aspires to revolutionize the typical voter's perception of civic engagement and ultimately elevate both the quantity and quality of community involvement. At its core, LD is predicated on empowering voters to determine their mode of participation. This can be achieved by either casting a vote directly, as in direct democracy, or by entrusting a proxy to act on their behalf, as in representative democracy. Notably, delegations are transitive, meaning that a delegate's vote can be delegated afresh, and at the end of the day a voter that has decided to cast a ballot, votes with a weight dependent on the number of agents that she represents, herself included. As a result of its flexibility, Liquid Democracy is alleged to reconcile the appeal of direct democracy with the practicality of representative democracy, yielding the best of both worlds. The origin of the "liquid" metaphor remains a matter of debate up to date, with one view being that it stems from the ability of votes to flow along delegation paths, while an alternative view argues that it arises from the ability of voters to revoke delegation approvals and continuously adjust their choices. At what follows, we will dive into how the second opinion serves as a significant driving force behind our research, providing the essential impetus for our endeavors.

According to [BZ16] there is a number of features that suffice to establish a framework as a Liquid Democracy one. Most of them are related to the transitivity property and to the options given to the voters about casting a ballot or choosing representatives. These are more or less taken into account in all relevant works that come from the field of Computational Social Choice. A further aspect, called *Instant Recall*, encompasses the ability of voters to withdraw their delegation at any time. As a matter of fact, in practice, elections allow for extended (sometimes structured) periods of deliberation, until the votes are finalized, and Liquid Democracy could serve as a means of debate empowerment. A withdrawal of delegation may occur for a variety of reasons: a voter may develop doubts on the integrity of her existing representative, or she may discover a higher alignment of opinion with a different representative, or she may simply obtain a better understanding of the election issue under consideration. A characteristic that is being shared by all the works in the AI community is that they all seem to ignore the Instant Recall feature, and examine isolated static delegation profiles. This oversight was identified and criticized by the team behind the LiquidFeedback platform [BKN+14], the most influential and large scale experiment of LD. In [BKN+22], inter alia, they state:

[BZ16] Blum and Zuber (2016): Liquid Democracy: Potentials, Problems, and Perspectives.

[BKN+14] Behrens et al. (2014): The Principles of LiquidFeedback.

[BKN+22] Behrens et al. (2022): The Temporal Dimension in the Analysis of Liquid Democracy Delegation Graphs.

> *"In a governance system with a continuous stream of decisions, we expect that participants observe the actions (and even non-actions) of other participants, in particular the activities of their (direct and indirect) delegates as well as the activities of other participants. Based on their observations, we expect participants to adapt their own behaviour in respect to setting, changing, and removing delegations and their own participation. Based on the track records of the participants, a network of trust or dynamic scheme of representation proves itself to be a responsible power structure. [...] We believe that the effects that occur through observation and adaptation over time are an essential prerequisite for a comprehensive understanding of Liquid Democracy, (which) requires a broader view, namely adding a temporal dimension to delegation models."*

Leaving aside the lack of temporal aspects in the literature, there are also additional concerns to address in traditional LD models. A crucial disadvantage is that we may experience delegation cycles or delegation paths towards abstainers, which result to inevitably lost votes. A way that has been suggested in theory [BDG+22; GKM+21; KR20] and has been implemented in practice [HL15], in order to mitigate such issues is to allow each delegating agent to specify an entire set of agents she approves as potential representatives together with a ranking that indicates her preferences. Nevertheless, even with these efforts, the discussed issues may still arise at the election-day. And here is where the temporal dimension can come into play! The main focus of our work is in proposing a framework that leverages temporal information to address such concerns, while also providing a valuable tool for deliberation. In the realm between the algorithmic and axiomatic approaches, our study examines the existence of efficient *delegation rules*, i.e., centralized algorithms that take as input the information from the deliberation phase and prescribe for each non-abstaining participant, a delegation path to a voter who casts a ballot, so as to meet the axioms outlined below (for formal definitions and further elaboration refer to Section 6.1):

Confluent Delegation Rules. In models incorporating multiple, ranked, delegations, as the one under consideration, an esteemed property is *confluence*, which posits that each voter should have at most one other immediate representative in the final outcome [BDG+22]. This desirable attribute guarantees that every voter is instructed to take a single action among the three options: vote, abstain, or delegate her own and all received ballots to a specific voter. On the contrary, a non-confluent rule may prompt a voter to delegate different ballots received from different voters to different representatives, or fractionally distribute her acquired ballots. Such suggestions can be challenging for a voter to follow (even though they may indeed be meaningful in certain occasions). In addition to its intuitive nature, confluence is also significant for maintaining transparency and preserving the high level of accountability inherent in classic LD [GKM+21].

[BDG+22] Brill et al. (2022): Liquid Democracy with Ranked Delegations.

[GKM+21] Gölz et al. (2021): The Fluid Mechanics of Liquid Democracy.

Time-Conscious Delegation Rules. The necessity for incorporating the temporal dimension in the design of delegation rules becomes evident when considering an election where the ballots at the very end of the deliberation phase produce a cycle or a path to an abstainer, for some voter(s). This, unquestionably problematic, scenario has been a widely recognized issue in the literature, that results in ballot loss. Our main insight proposes that a viable compromise to address such situations is by looking into approvals expressed during the previous time-steps of the deliberation phase. Our work operates under the premise that if a voter v decides to trust a voter u, at a given time t, then v accepts any decisions made by u at time t or earlier (up to a certain number of time-steps prior to t, which could be given as a parameter by voter v). This is because the decision to approve a delegation to u is based on what v observes in the previous time-steps and up until time t. However, voter v still retains the right to revoke her approval to u at a later point in time. If this occurs, then u is permitted to finally represent v only if she chooses an action that she had declared at or before time t and, consequently, u will represent v with a specific opinion that v had indeed approved. We refer to the rules that produce delegation paths respecting in such a way the ordering of the time-instants at which a delegation is made available, as *time-conscious*. We highlight that, in instances where phenomena of ballot loss do not appear at the end of the deliberation phase, the solutions we suggest need not involve delegations made in previous time-steps; for the remaining instances avoiding ballot losses is impossible without utilizing past delegations. Hence, one can view our procedures as unavoidable compromises that strike a balance between voters' participation and satisfiability.

Contribution.

Conceptually, the main contribution of this chapter lies in explicitly incorporating a temporal dimension into an existing election framework, broadening the solution space to alleviate recognized drawbacks. This is putting a stake in the ground in bridging a significant research gap identified by practitioners. From a technical perspective, our work is making a pioneering contribution to the Computational Social Choice literature, since our study incorporates concepts and techniques from temporal graph theory: a notably novel approach in the field.

We study the compatibility of computational tractability with desirable axioms of delegation rules, and with the objective of reducing the loss of votes. More specifically, we are interested in polynomially computable rules that maximize the total utility of the electorate and are also time-conscious and confluent. Unfortunately, despite the natural appeal of the studied requirements, it turns out that this is too much to ask for: our main result establishes that such a delegation rule does not exist, unless P=NP, even for simple of the model. Therefore, the best one could hope for is to sacrifice one of the considered axioms or to resort to special

cases of the problem. We present results in both directions that effectively circumvent the hardness. We show that while dropping the requirement of time-consciousness is sufficient for designing an efficient delegation rule, certain instances of the problem remain hard when we give away confluence. In response, we introduce some natural restrictions under which positive results emerge in the latter case. Furthermore, when we insist on both axioms, we describe a rule that significantly outperforms the brute-force solution by exhibiting a running time that is exponentially dependent only on the number of delegating voters.

Related Work.

To offer a comprehensive view for readers unfamiliar with Liquid Democracy settings, we begin by providing a concise historical context for the field. This involves a discussion of works related to LD, aligning with the concept of "beginning at the beginning" akin to the advice in "Alice's Adventures in Wonderland"; the author of that novel, Charles Dodgson (also known by his pen name Lewis Carroll), as early as 1884 [Car84], considers an idea that was meant to be of vital importance for what we call Liquid Democracy today. According to [Beh17], it seems that he is the one that, before all else, discussed the aspect of giving the agents the power to transfer to others their acquired votes. Later on, it was Gordon Tullock [Tul67] who initiated the discussion about models that aspire to occupy the ground between direct and representative democracy, by suggesting a model that allows voters to decide whether they are interested in casting a ballot or delegate to another voter. Shortly after, unlike Tullock's suggestion, James Miller [Mil69], brought forward the idea that voters should not only choose their mode of participation but should also enjoy the ability to retract a previously given delegation in a day-to-day basis. At what concerns the nomenclature of LD, the precise origins are unknown. The best one could refer to, is its seemingly first [Pau14] recorded appearance (in an obsolete wiki, preserved only on the Internet Archive [say04a; say04b]), in which a user nicknamed "sayke" discoursed about a voting system that lies between direct and representative democracy and aims at increasing civic engagement. However, none of these sources discussed explicitly the aspect of transitivity of votes, as Dodgson did. Reinventions, amendments and compositions of these ideas started to appear at the early 00's and we refer to [For14] for an overview of them. The earliest published works that incorporate the aspects of LD, (roughly) as we consider it today are [BBC+11; CMM+17; For02; Gre15]. Nowadays, Liquid Democracy is one of the most active research areas in Computational Social Choice [Bri18; Pau20].

As already mentioned, the primary motivation of our work is due to [BKN+22]. The framework we suggest is a generalization of the model in [BDG+22]. Furthermore, our optimization objective coincides with the one in [EGP19; MP21a].

[BKN+22] Behrens et al. (2022): The Temporal Dimension in the Analysis of Liquid Democracy Delegation Graphs

[BDG+22] Brill et al. (2022): Liquid Democracy with Ranked Delegations

[EGP19] Escoffier et al. (2019): The Convergence of Iterative Delegations in Liquid Democracy in a Social Network

[MP21a] Markakis and Papasotiropoulos (2021): An Approval-Based Model for Single-Step Liquid Democracy

To our knowledge, our work is the first that incorporates temporal aspects in LD models, towards the avoidance of ballot loss. Models exploring dynamic aspects in LD have also been considered in [EGP19] and [BGL19], but from a different angle, namely with a focus on a game-theoretic perspective. To be more precise, our work introduces a temporal dimension by explicitly considering preferences that voters hold before the election day. This distinguishes it from [BGL19; EGP19] where instead of deliberation, they iteratively execute the ballot-casting phase towards reaching "stable" states, rendering past voters' choices completely unacceptable for the final outcome. Many other directions and questions related to LD have been extensively examined in the recent literature: indicatively, recently published works explored aspects including the study of voting power concentration through the lens of parameterized complexity [DMS21], the efficiency of altering delegations to achieve consistency in participatory budgeting settings [JST22], the application of power indices and criticality analysis to voters [CDG23], election bribery in a LD setting [BBR+22], and the evaluation of LD's epistemic performance [RHB+22].

6.1 Election Framework and Definitions

In the current section, we provide a detailed description of Temporal Liquid Democracy, the time-aware voting formalism that we study. We consider elections in which a set V of n voters should reach a decision on a certain issue. Apart from voting themselves, the participants are given two additional options: abstaining or delegating to other voters. The voters also have some time available to consider what to do (e.g. to get informed on the issue at hand or to observe other voters' choices) and they are allowed to change their mind, perhaps multiple times, until the actual election-day. We say that such an election is a *Temporal Liquid Democracy Election* (t-LD in short) if it consists of:

➛ A *deliberation phase* of L rounds, where, each voter v, tries to decide whether to personally vote or not. If she decides to cast a ballot at some time $t \in [L]$, we consider this as her final decision that will not change in the remaining rounds. Otherwise, at every time-instant $t \in [L]$, and as long as a voter v has not decided to cast a ballot herself, she is asked to specify the following:

  - A set of approved voters $S_v^t \subseteq V \setminus \{v\}$ (which may be the empty set, if v wants to abstain at round t), indicating the voters that she trusts to cast a ballot on her behalf, possibly with different levels of confidence. These voters may in turn also be willing to delegate their vote to others.
  - A (weak) preference ranking over the voters in S_v^t, which induces a partition of S_v^t into preference groups, according to v. This is accompanied by a positive integer score $sc_v^t(i)$, indicating the utility or happiness level that v experiences[1] if a voter from her i-th most preferred group at time t, will ultimately be selected as her immediate representative.[2]

[1] Scores are implicitly assumed to be comparable not only between voters but also across different time-instants for a given voter; the rationale behind this will become apparent shortly.

[2] It is a fact that only soliciting scores (excluding rankings) from voters is sufficient. However, for simplicity, we could even allow voters to only submit rankings (excluding scores). Therefore, this presentation choice aims to facilitate adaptation to different model variants.

  - A non-negative integer-valued trust-horizon parameter $\delta_v^t \leq t-1$, by which, she indicates approval for the views held by any voter in S_v^t up to δ_v^t time-steps prior to time-instant t.

➛ A *casting phase*, in which all the voters that, during the deliberation phase, expressed willingness to vote (and only these), cast a ballot on the issue under consideration. Every voter who did not declare an intention to abstain at $t = L$, or to vote, is assigned a representative through a prespecified delegation rule that takes into account the entire deliberation phase. The winner(s) of the election then are elected using some weighted voting rule, where the weight of a voter is based on the number of participants that she represents.

For an illustrative exposition of our model we refer to Example 6.1 (provided at the end of the section as some required notation and terminology are yet to be introduced). We now elaborate on the input that is required from the voters, during the deliberation phase. The preference ranking facilitates voters to express different levels of confidence towards other participants who could potentially represent them. Also, the scoring function allows the model to capture the cases where a voter is willing to either increase her scores over rounds due to becoming gradually more informed about another voter's opinion or in the opposite direction, decrease scores due to becoming more hesitant about who represents her. Realistically, we expect voters to have just a few preference groups, and hence they do not need to submit too many numerical parameters. Furthermore, the intuition behind the trust-horizon parameters is that the decision of a voter v to approve a delegation to u at time t, can be based only on looking at the behavior of u in the previous rounds and up until time t of the deliberation phase. Since a voter v may not agree with u in all previous time-steps, the parameter δ_v^t specifies that v agrees with the choices made by u at any preceding time that is no more than δ_v^t time-instants before t. Here by the choices of u, we mean either that u chooses to vote herself or she chooses to further delegate to other voters with her own trust-horizon. A simple case to have in mind for the trust-horizon parameter is when $\delta_v^t = t-1$ (i.e., v trusts whatever u has chosen at any time in the past). If this holds for every voter v and for any $t \in [L]$, we say that the election profile is of *retrospective trust*.

Variants of the Model and Practical Considerations.

At what follows we (a) present generalizations of our framework where our positive results persist, (b) identify special cases of the studied problem, that are of particular interest, where our negative results remain valid and (c) explore practical adaptations of the model to enhance its applicability in real-world scenarios.

We commence by listing certain natural generalizations of the studied framework for which the algorithms we proposed still work.

- It would be meaningful to allow a voter to specify a different time-horizon parameter for each of her approved representatives. Notably, our findings are not impacted by the assumption of a uniform trust-horizon for approved representatives.
- We have assumed that once a voter commits to voting, her intention to cast a ballot remains unchanged until the election-day. This assumption is justified by the fact that once a voter has committed to becoming more informed on the topic, participating in further deliberation is deemed redundant. Nonetheless, this assumption is made only for technical convenience and could be dropped.
- Our work reflects the premise that "it is better for a voter to be represented by someone she previously approved (even if the approval is currently removed) than losing her ballot (due to delegation cycles or paths to abstainers[3] at $t = L$)". One way to interpret this, is that it captures scenarios in which voters make adjustments rather than complete revocations of their preferences, i.e., removing a delegation may mean that a voter found more preferable choices but not that she necessarily disagrees with past delegations she declared. Nevertheless, we emphasize that this interpretation is not essential for our results; in cases where a voter wants to fully revoke a past approval (perhaps due to new information), this can be accommodated through simple modifications, e.g., a final round allowing to specify the complete removal of some past approvals from any consideration.

We now shift our focus towards exploring noteworthy subclasses of the studied problem. Remarkably, both hardness results of Theorem 6.1 and Theorem 6.3 continue to stand firm in these cases with minimal or no adjustments in their proofs.

- It is reasonable to consider a scenario in which voters' utility is based on the recentness of approval, i.e., each delegating voter prioritizes the selection of a representative that has been approved by her as late as possible in the deliberation phase.
- We acknowledge the asymmetry between abstainers' and non-abstainers' past opinions that stems from viewing abstention as a complete denial of participation, potentially unsuitable for subsequent delegation. While this motivation may be questioned, crucially, we underline that both of our negative results hold even if abstainers' past approvals are not to be disregarded.
- Another asymmetry appears between abstaining and casting voters. It's worth mentioning that abstaining voters may also finalize their decision sooner in the process, akin to the decision of casting voters.

[3] Paths to abstainers represent a recognized downside of LD, nevertheless, our model readily accommodates a suggestion of allowing such paths, as our results still hold when seeking solutions that only prohibit delegation cycles: a well-motivated and undeniable desideratum.

Although our aim is to examine the model in its fullest generality, we stress that in potential real-life implementations, the voters may not need to submit all the information that we have described in every round. This adjustment contributes to enhancing the framework's accessibility and usability. Considerations regarding the adaptability and practical applicability of our model are expounded below.

- ➛ The scoring function could be automatically generated by the system, given the (weak) ranking on S_v^t submitted by each voter. For instance, one could use the Borda-scoring function (as in [BDG+22]), under which, at any time-instant, a voter assigns a score of 1 to her last preference group, a score of 2 to her second to last group, etc, or any other appropriate method. It is evident now that our model is a strict generalization of the model considered in [BDG+22], not only because of the temporal dimension but also because of the more general scoring functions that we allow.
- ➛ The trust-horizon parameter could also be prespecified, so that the voters do not need to submit any information regarding it, either by assuming that the trust of every voter goes arbitrarily back in time or for a fixed number of steps prior to each approval.
- ➛ If voters have the same preferences for consecutive time-steps, they would not need to re-specify them in every time-step t.

Delegation Rules. In the elections we consider, we have three types of participants. We refer to the voters that declared intention to vote as *casting voters*, and these are the only voters who will indeed cast a ballot at the election-day. The non-casting voters that will abstain from the election are precisely those who do not approve anyone at the final time-step, e.g. a voter v such that $S_v^L = \emptyset$. We refer to such voters as *abstaining voters*. The rest of the voters will be called *delegating voters*. As evident from the introduction of current chapter, and as will be further illustrated by the example at the end of this section, the temporal dimension could be considered valuable, via the notion of time-consciousness, when the examination of the isolated instance at $t = L$ cannot produce a feasible solution (i.e. delegation cycles or paths towards abstainers are unavoidable). A delegation rule is a mechanism that "resolves delegations" to address such problematic cases, i.e., a procedure that ultimately assigns to each delegating voter, a casting voter, possibly via following some path of trust relationships. More formally, a delegation rule is a function that takes as input the voters' preferences, as reported during the deliberation phase of a t-LD election, and outputs a path to a casting voter, for every delegating voter. A delegation rule should ask casting voters to vote, abstaining voters to abstain and should not suggest any delegation path towards an abstainer or introduce delegation cycles.

Temporal Graphs. The driving force in our work is to model and analyze t-LD elections using principles from temporal graph theory. We start with a basic overview of the concept and the terminology of temporal graphs and following this, we will introduce some notation that we will use in the remainder. In high level, a temporal graph is nothing more than a simple, called *static*, graph in which a temporal dimension is being added, i.e., a graph that may change over time. Frequently, a temporal (multi)graph is being expressed as a time-based sequence of static graphs. For convenience, we will use an equivalent definition, under which, a (directed) temporal (multi)graph $G(V, E, \tau, L)$ is determined by a set of vertices V, a (multi)set of directed, temporal edges E, a discrete time-labelling function τ that maps every edge of E to a non-empty subinterval of $[1, L]$, and a lifespan $L \in \mathbb{N}$. If the edges of E are weighted according to a function $w : E \to \mathbb{N}$, then we say that G is weighted. The interval $\tau(e) = [s_e, t_e]$, that is assigned to an edge e, indicates that e is available at the time-instants that belong to $\tau(e)$ (it is possible also that, $s_e = t_e$). By allowing G to be a multigraph[4] it is permitted for an edge to be present in multiple (disjoint) time-intervals. Unless otherwise stated, henceforth, by the term *graph*, we denote a weighted directed temporal multigraph. For more details on temporal graphs we refer to a relevant survey [Mic16], as well as to the fundamental and influential works [KKK02; Kos09; MMS19]. The *static variant* of a temporal graph is the static graph that emerges if we ignore the time-labels of its edges. We call a (temporal) graph *temporal directed tree rooted at vertex* r if its static variant contains a directed path towards r from every other vertex and its undirected variant is a tree. A crucial concept for our work, in the context of temporal graphs, is the notion of time-conscious paths, that satisfy a monotonicity property regarding the temporal dimension of their edges. Consider a temporal graph $G(V, E, \tau, L)$, coupled with a tuple $\delta_v = (\delta_v^t)_{t \in [L]} \in \mathbb{N}^{[L]}$ for every vertex v of V. Let also $\delta = (\delta_v)_{v \in V}$ and for notational convenience we will occasionally include δ in the description of a temporal graph, denoted then by $G(V, E, \tau, L, \delta)$. We say that a path in G from v_1 to v_{k+1} is δ-*time-conscious* if it can be expressed as an alternating sequence of vertices and temporal edges $(v_i, (e_i, t_i), v_{i+1})_{i \in [k]}$, such that for every $i \in [k]$ it holds that $e_i = (v_i, v_{i+1}) \in E$, $t_i \in \tau(e_i)$ and for every $i \in [k-1]$ it holds that $t_i \geq t_{i+1} \geq t_i - \delta_{v_i}^{t_i}$. Hence, time-consciousness specifies a traversal of an edge e at a time that is no later than the previous edge in the path, and while e is present.[5] Similar notions have been applied to various domains including flight connections detection [WCK+16], information diffusion [HFL15] and disease control through contact tracing [BHN+20]. In the remainder of Section 6.1, it will become more clear how this notion fits in our framework. We also call δ-time-conscious, a temporal directed tree, rooted at a vertex r, if all its paths towards r are δ-time-conscious. For illustrative examples of the terminology refer to Section 6.A.

Modelling t-LD Elections as Temporal Graphs. The deliberation phase of a t-LD election, can be modeled as a weighted directed temporal multigraph $G(V \cup \{\veebar\}, E, \tau, L, w, \delta)$, that is formed by

[4] We are using multigraphs instead of (simple) graphs merely for technical convenience, and we note that, alternatively, one could work with graphs by letting τ be a function that maps edges to a set of subintervals of $[1, L]$.

[Mic16] Michail (2016): An Introduction to Temporal Graphs: An Algorithmic Perspective.

[KKK02] Kempe et al. (2002): Connectivity and Inference Problems for Temporal Networks.

[Kos09] Kostakos (2009): Temporal Graphs.

[MMS19] Mertzios et al. (2019): Temporal Network Optimization Subject to Connectivity Constraints.

[5] It is crucial to draw attention to a departure from the norm, which is intentionally chosen to suit the requirements of our application: the definition of δ-time-conscious paths deviates from the traditional understanding of (temporal) paths in temporal graph theory, where edges are traversed in a later order. We refer to the proof of Theorem 6.1 for a comprehensive discussion.

- a vertex in V for every voter, as well as a special vertex $\veebar$ representing the voters' commitment to cast a ballot,
- a multiset E of temporal edges that contains the following: (i) edges that represent the approvals for delegation per round via a function τ that assigns a time-label to every edge, (ii) an edge $(v, \veebar)$, for which $L \in \tau((v, \veebar))$, for every vertex v that corresponds to a casting voter,
- a lifespan L that represents the duration of the deliberation phase,
- a function w that assigns a weight to every edge (v, u) of E, according to sc_v^t, provided that $t \in \tau((v, u))$,
- a vector δ that, for every voter v, contains a tuple $(\delta_v^t)_{t \in [L]}$, as declared by v during the deliberation phase.[6]

If C is the set of casting voters, then if any such voter had indicated preferences for potential representatives before deciding to cast a ballot, these preferences, and their corresponding edges, can be safely disregarded. More precisely, only the following two types of edges may exist: directed edges of the form $e = (v, u)$ for $v \in V \setminus C$ and $u \in V$ with $\tau(e) = [s_e, t_e]$, indicating that at any time-instant $t \in [s_e, t_e]$, voter u belongs to S_v^t, and directed edges $e = (v, \veebar)$ for $v \in C$ with $\tau(e) = [s_e, L]$, indicating that from time s_e and onwards, voter v agrees to cast a ballot. As already explained before, the deliberation phase implicitly partitions the set of voters V into three sets: the set of casting voters C, the set of abstaining voters A and the set of delegating voters D. More formally, $C = \{v \in V : (v, \veebar) \in E\}$, $A = \{v \in V \setminus C : L \notin \tau((v, u)), \text{for any } (v, u) \in E\}$ and $D = V \setminus (C \cup A)$. The weight function w indicates the cardinal preferences of a voter, as implied by the scores that accompany her preference rankings during the deliberation phase. Additionally, for convenience, we set to zero the weights of edges (v, u) such that v corresponds to a casting or an abstaining voter. This choice can be justified by the upcoming discussion of the optimization objective in the "Electorate's Satisfaction" paragraph. Given a graph $G(V \cup \{\veebar\}, E, \tau, L, w, \delta)$ that models a t-LD election, a delegation rule returns, for every delegating voter v, a weighted directed temporal path from v to $\veebar$, which infers an assignment of every delegating voter to a casting one. We call a delegation rule *efficient* if its output can be computed in polynomial time.

Our Target Axioms Under the Temporal Framework. We proceed to define, within the framework of temporal graphs, the axioms that constitute the core focus of our work. We have already discussed them in the analogous paragraphs of the introduction of the current chapter (see page 106) and we now formally define them using the framework of temporal graphs.

Definition 6.1 *Let $G(V \cup \{\veebar\}, E, \tau, L, w, \delta)$ be a graph modelling a t-LD election.*

[6] For convenience, we allow δ to have empty entries, corresponding to casting voters or to time-steps during which the corresponding voter abstained.

1. *A delegation rule is time-conscious, if for every delegating voter v, the delegation path output for v is a δ-time-conscious directed temporal path.*
2. *A delegation rule is confluent, if the union of the paths output for all the delegating voters is a directed temporal tree, rooted at vertex $\underline{\vee}$, that spans the vertices of $V \setminus A$.*

The definition of time-consciousness guarantees that all paths suggested by the delegation rule satisfy the constraints imposed by the voters, regarding their trust-horizon parameters. Hence, for any edge (v, u) in an output path, u must perform an action (i.e., choose an edge corresponding to a further delegation or vote directly) that she had declared at a time that was approved by v. The definition of confluence guarantees that for every delegating voter v, there is a unique path to a casting voter, that is intended to serve v and voters who delegated to v.

In Section 6.A and in Example 6.1 we provide some examples of time-conscious and confluent solutions, for further illustration.

Electorate's Satisfaction. We make the usual assumptions for Liquid Democracy models that (a) voters completely trust their representatives and (b) trust between voters is transitive. This implies that if voter v accepts voter u as her potential representative, she concurs with any subsequent choice made by u and also extends trust to any voter w who may be entrusted by u. Hence, we note that the utility experienced by a delegating voter from a delegation rule can be considered as a local one, being contingent solely on the voter's immediate representative and not influenced by further choices made by the chosen representative. Therefore, the utility of a delegating voter can be determined by the score that she declared for her immediate representative, specified by the delegation rule. Note that two different time-instants t, t' may exist such that $u \in S_v^t \cap S_v^{t'}$. In these cases, given that the output of a delegation rule is a set of temporal paths, if the rule suggests a delegation from v to u, it also explicitly specifies the time-instant at which the delegation will occur, say e.g. at time t' and, thereby the utility of v is equal to $sc_v^{t'}(i)$, if u belongs to the i-th most preferred group of v, at time t'. Regarding now the casting voters, we do not take into account their utility since their will to cast a ballot has been realized; we do the same for abstaining voters. We consider as infeasible every solution that asks a casting (resp. abstaining) voter to delegate her ballot or abstain (resp. vote), and hence, our focus will be on the welfare of the delegating voters. Finally, the quality of a rule is assessed by the total satisfaction it elicits from the electorate which is expressed as the sum of utilities of all delegating voters. As extensively discussed already, our goal is to consider the entire deliberation phase so as to address instances where it is unattainable to achieve feasibility by looking only at the final time-step. Nevertheless, the proposed framework also facilitates exploration of a broader setting, wherein allowing past consultation can improve a solution's quality (with respect to the electorate's welfare) even when feasible solutions at the final time-step do exist. Our optimization objective is to maximize the electorate's satisfaction, a natural and classic choice not only for Liquid Democracy setups but also for Computational Social Choice in general. The algorithmic problem our work focuses on is formally defined below.

RESOLVE-DELEGATION

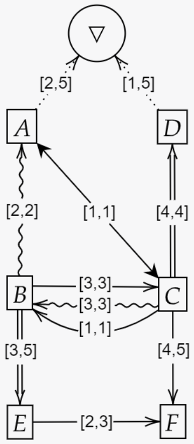
Given: A graph $G(V \cup \{\underline{\vee}\}, E, \tau, L, w, \delta)$ that represents the deliberation phase of a t-LD election.

Output: Compute a weighted directed temporal path from each delegating voter to $\underline{\vee}$, with the aim of maximizing the total utility derived from the delegating voters, defined as the sum of the weights of the paths' first edges.

Example 6.1 As an illustration, consider the following instance of a t-LD election with 5 rounds and 6 voters, namely Alice, Bob, Charlie, Daisy, Elsa, and Fred. Their preferences are outlined below:

- Alice initially intended to delegate to Charlie. In the second round, she decided to get informed about the considered issue and vote.
- Bob did not participate in the deliberation phase during the first round, but approved Alice in the second round. In the third round, he revoked his approval of Alice and instead approved Chris and Elsa. Bob's approval of Elsa remained until the final round.
- Charlie approved Alice only in the beginning of the election. He also approved Bob in the first and third round, but removed his approval (and abstained) in the second round. In the fourth round, Charlie approved both Daisy's and Fred's perspective on the topic at hand, but he removed his approval of Daisy in the final round.
- Daisy expressed interest in being a casting voter from the beginning until the end of the deliberation phase.
- Although Elsa intended to delegate her ballot to Fred at rounds 2 and 3, ultimately both refrained from participating.

The described instance can be visualized using the graph shown below.



We assume that $\delta_B^t = 1$ and $\delta_C^t = t - 1$ for every $t \in \{1, 2, \dots, 5\}$. The scores assigned by the voters to their approved representatives are encoded by the form of the edges, where curly edges have weight 1, straight edges have weight 2, and double-lined edges have weight 3. Dotted edges indicate the casting voters. The labels of the edges represent the time-intervals of their presence. In this instance, Alice and Daisy form the set of casting voters, while Elsa and Fred abstain. Therefore, edge (A, C) can be removed, since Alice will definitely cast a ballot. In a δ-time conscious solution, Bob would not delegate to Charlie, since no δ-time-conscious path to $\veebar$ using the edge (B, C) exists, for instance, edge (C, A) violates the time-horizon declared by Bob. Similarly, Charlie would not delegate neither to Alice nor to Bob at time 1. Since we do not allow Bob to delegate to an abstainer, he must delegate to Alice, whom she trusted at time 2. Then, there are two possible outcomes for the delegation rule, depending on the choice made for Charlie. The edge that maximizes Charlie's utility is (C, D). Therefore, the optimal delegation rule that is both time-conscious and confluent, would suggest the set of paths $\{((C, D), (D, \veebar)), ((B, A), (A, \veebar))\}$, achieving a total satisfaction score of 4. Finally, in this example it is plainly evident how the temporal dimension comes to the rescue: if one were to focus solely on the snapshot taken at time 5, disregarding the information garnered from the deliberation phase, the only option would be to ask Bob and Charlie to delegate to abstaining voters since at time 5 they only approve Elsa and Fred respectively. Instead, our framework utilizes the information obtained throughout the deliberation phase to propose an outcome that avoids paths towards abstainers and also avoids delegating cycles.

6.2 Algorithms and Hardness Results

In this section we explore the compatibility of the axioms we have put forward from Section 6.1, with efficient computation.

Our first result shows that it is impossible to have polynomially computable utility maximizing delegation rules that satisfy simultaneously the axioms of time-consciousness and confluence, unless P=NP, even under simple, natural restrictions. Before stating the result, we discuss the types of instances for which we establish hardness. It is expected that in real-life elections, voters tend to exhibit a relatively stable opinion over time, and do not revise their preferences numerous times during the deliberation phase, due to the required effort to process new information. Similarly, it is reasonable to expect that due to limited cognitive capacity, the voters are only able to partition their accepted representatives into a few disjoint preference groups. The theorem that follows demonstrates that the computational intractability of RESOLVE-DELEGATION persists even when we limit the voters to changing their mind at most once during the deliberation phase and partitioning their accepted representatives into at most two groups, at each round. Furthermore, it holds even for instances of retrospective trust, and with Borda-scoring functions. Therefore, the primary takeaway is that incorporating temporal aspects in conjunction with natural requirements does come at a computational cost.

Theorem 6.1 *RESOLVE-DELEGATION in a time-conscious and confluent manner is NP-hard, even for profiles of retrospective trust and under the Borda-scoring function.*

Proof. Given a graph $G(V \cup \{\underline{\vee}\}, E, \tau, L, w, \delta)$ and a parameter k, we call Π the decision variant of RESOLVE-DELEGATION in a time-conscious and confluent manner, which asks for the existence of a solution with total satisfaction at least k. At what follows, we provide a reduction to Π from the NP-hard problem [HFL15] MINIMUM TEMPORAL SPANNING TREE (t-MST).

[HFL15] Huang et al. (2015): Minimum Spanning Trees in Temporal Graphs.

We note that in temporal graph theory the term *time-respecting* is being used to describe, a temporal path $(v_i, (e_i, t_i), v_{i+1})_{i \in [\ell]}$, such that for every $i \in [\ell]$, it holds that $e_i = (v_i, v_{i+1})$, $t_i \in \tau(e_i)$, and $1 \leq t_1 \leq t_2 \leq \cdots \leq t_\ell \leq L$ (also called "journey" or simply "temporal" in the related literature). The difference between time-respecting and δ-time-conscious paths is that the paths of the former type are formed by edges whose time-stamps are in non-decreasing order of visiting, in contrast to the paths of the latter type, whose edges have time-stamps in non-increasing order, and on top of that, satisfy a waiting-time constraint indicated by vector δ. We also refer to Section 6.A for an example.

In the t-MST problem, we are given a temporal graph $G'(V', E', \tau', L', w')$, as well as a root vertex $u_0' \in V'$ and an integer k'. We are asked for a directed temporal tree of G', called T', of edge set E'', that spans the vertices of V' and that has a time-respecting path from u_0' to every vertex of V', such that $\sum_{e \in E''} w'(e) \leq k'$.

It is important to highlight that t-MST is NP-hard even for the case where $w'(e) \in \{1, 2\}, \forall e \in E'$, and for every $v \in V'$ there exists a $u \in V'$, such that $L' \in \tau'((u, v))$. It is without loss of generality to assume that u_0' has no in-coming edges in E'. Furthermore, the hardness holds for instances in which for any pair of vertices u, v of the input graph $G'(V', E', \tau', L', w')$, either $(u, v) \notin E'$, or there are two copies, e_1 and e_2 of (u, v) in the multiset E'. In the second case, it also holds that $\tau'(e_1) = [1, L' - 1], \tau'(e_2) = [L', L']$ and that $w'(e_1) = 2, w'(e_2) = 1$.

Given such an instance $(G'(V', E', \tau', L', w'), u_0', k')$ of MINIMUM TEMPORAL SPANNING TREE we create an instance $(G(V \cup \{\underline{\vee}\}, E, \tau, L, w, \delta), k)$ of Π as follows:

- let $L = L'$,
- for every vertex $u' \in V'$ we add a vertex $u \in V$,
- for every directed edge $(u', v') \in E'$ we add a directed edge (v, u) such that $w(v, u) = 3 - w'(u', v')$ (recall that $w'(u', v') \in \{1, 2\}$) and $\tau((v, u)) = \tau'((u', v'))$,

- we add a special vertex $\veebar$ and a directed edge $e = (u_0, \veebar)$ such that $w(e) = 0$ and $\tau(e) = [1, L]$,
- we add one more special vertex $a \in V$,
- for every $t \in [L]$ and $v \in V$ such that there exists in E an out-going edge from v at time t of weight 2 but not of weight 1, we add an edge (called "dummy") $e = (v, a)$ such that $w(e) = 1$ and $\tau(e) = [t, t]$,
- for every vertex $v \in V \setminus \{a\}$ that corresponds to a non-casting voter and for every $t \in [L]$, we set $\delta_v^t = t - 1$,
- we set k to be $3(n - 1) - k'$.

The special cases for which the hardness holds, stated in the statement of the theorem, simply follow by the construction. We make the following observations regarding the t-LD election represented by G.

- The only vertex that has an out-going edge to $\veebar$ is u_0. Such an edge is available at the final time-instant L and thus the voter that corresponds to u_0 agrees to cast a ballot till the end of the election; which makes her the only casting voter.
- The only vertex that doesn't have an out-going edge at time L is a. More precisely, a has no out-going edges during $[1, L]$ and thus the corresponding voter opts to abstain from the beginning until the end of the election; which makes her the only abstaining voter.
- The weights of the out-going edges of every vertex v of $V \setminus \{\veebar, u_0\}$ at any time-instant t, indeed express a weak ranking over the voters that are being approved by v at time t, due to the dummy edges.

Before continuing, we observe that vertex a as well as the edges towards a do not affect the rest of the reduction since such edges do not belong to any path to $\veebar$, and are not part of any feasible solution of Π. Hence, it is safe to focus on the subgraph of G induced by $V \cup \{\veebar\} \setminus \{a\}$, which, for simplicity, will be called $G(V \cup \{\veebar\}, E, \tau, L, w, \delta)$, from now on.

For the forward direction, say that (G', u_0', k') is a YES-instance of t-MST having T' as a certificate. We will prove that (G, k) is also a YES-instance of Π. We select an arbitrary path p' of T' that has u_0' as its source vertex, and we rename its vertices and edges so as $p' = (u_{i-1}', (e_i', t_i), u_i')_{i \in [q]}$, for some $q \in \{1, 2, \dots, n\}$. Since T' is a subgraph of G', the existence in p' of the edge $e_i' = (u_{i-1}', u_i')$, for $i \in [q]$ and for which $t_i \in \tau(e_i')$ implies the existence of an edge $e_i = (u_i, u_{i-1})$ in G that is present at time t_i. Combining these edges we prove the existence of a path $p = (u_i, (e_i, t_i), u_{i-1})_{i \in [q]}$, in G. Since p' is time-respecting, it holds that $1 \leq t_{i-1} \leq t_i \leq L, i \in \{2, 3, \dots q\}$, and hence, given that the elections represented by G are of retrospective trust,

p' is a δ-time-conscious path. Finally, the path $p \cup (((u_0, \underline{\vee}), 1), \underline{\vee})$ is a δ-time-conscious path from u_q to $\underline{\vee}$. Combining such paths for every vertex $u_q \in V \setminus \{u_0\}$, we can create a subgraph T of G that is a δ-time-conscious tree rooted at $\underline{\vee}$ that spans the vertices of $V \cup \{\underline{\vee}\}$.

We now focus on the cost of the edges in T. The cost of all edges of T' is a sum of values $w'(e_i') \in \{1, 2\}$ and the number of edges in T' are exactly $n - 1$, where $n = |V'| = |V|$. Lets call $d_{T'}$ the number of edges of weight 2 in T' then $n - 1 + d_{T'} \leq k'$. Furthermore, it holds that for every edge of weight 2 (resp. 1) in T' there is an edge of weight 1 (resp. 2) in T and vice versa. Given that $w(u_0, \underline{\vee}) = 0$, the total weight of edges of T is

$$n - 1 + (n - 1 - d_{T'}) = 2(n - 1) + (n - 1) - k' \geq k.$$

For the reverse direction suppose that there is a directed temporal tree T that verifies a YES-solution of Π and say that E_T is its edge set. Since T is rooted at $\underline{\vee}$ and the only edge incident to it is $e = (u_0, \underline{\vee})$, then e is definitely part of T. Consider the graph T' that corresponds to the subgraph of G' that contains e as well as an edge (u', v') if and only if (v, u) belongs to $E_T \setminus \{e\}$. The fact that T' is a time-respecting directed temporal tree that spans the vertices of G' and has a path from u_0' to every vertex of G', follows by similar arguments to the forward direction of the proof.

We now need to prove that the total weight of the edges of T' is at most k'. It is known that the total weight of the edges of T is at least $k = 3(n-1) - k'$. Lets call d_T the number of edges of weight 2 in T, then $(n-1) + d_T \geq 3(n-1) - k'$. Since every edge of weight 2 (resp. 1) of G' corresponds to an edge of weight 1 (resp. 2) in G and vice versa, the total weight of the edges in T' equals

$$(n - 1) + (n - 1 - d_T) \leq 2(n - 1) + (n - 1 - 3(n - 1) + k') = k',$$

and this concludes the NP-hardness proof. □

We now explore roads to circumvent this impossibility result. Our proposal is to relinquish either the necessity for efficiency or one of the axioms of time-consciousness and confluence, in hopes of solving RESOLVE-DELEGATION. Our findings show that this strategy proves successful for some of the problems that emerge, which highlights that Theorem 6.1 is not devastating. Notably, most of the suggested procedures are simple enough and therefore are confirmed as strong contenders for practical applications.

We begin with studying the easiest variant of RESOLVE-DELEGATION in which the requirement of time-consciousness is being disregarded. This is mainly done for the sake of completeness since studying it requires overlooking the temporal dimension of the instance, which is the defining characteristic of our work. In order to solve efficiently RESOLVE-DELEGATION in a confluent but not necessarily time-conscious manner, the delegation rule can treat any input submitted by a voter at any time as if it was not subject to time-related constraints. Since confluence implies that the output should be a directed tree, and since the utility of each delegating voter is determined by its outgoing edge, then all edges of the tree with non-zero weight will contribute exactly once to the total satisfaction, and therefore, the objective is to find a (static) directed tree of maximum total weight, that is rooted at $\veebar$ and spans the vertices of $V \setminus A$. To solve this problem we leverage the well-known algorithm by Edmonds [Edm67] (also independently discovered in [Boc71; Chu65] and improved in [Tar77]) for the directed analog of the classic MINIMUM SPANNING TREE problem.[7] In this problem, given a weighted directed static graph $G(V, E, w)$ and a designated vertex $r \in V$, we are asked for a subgraph T of G, the undirected variant of which is a tree, of minimum total cost, such that every vertex of G is reachable from r by a directed path in T. It is important to note that in our case, the paths we need to compute are towards a fixed vertex, rather than originate from it. To apply Edmonds' algorithm, we adjust the graph G as indicated by Algorithm 2.

Algorithm 2: A confluent and efficient utility maximizing delegation rule for input $G(V \cup \{\veebar\}, E, \tau, L, w, \delta)$.

```
$G :=$ static variant of $G$
$V' := V \cup \{\veebar\} \setminus A$
$E' := \{(u, v) : (v, u) \in E \wedge v, u \in V'\}$
For every edge $e' \in E'$:
    $w'(e') := -w(e)$, where $e' \in E'$ corresponds to $e \in E$
remove duplicates from $E'$, retaining only the min-weight edge
let $G'$ be the (static) directed weighted graph $(V', E', w')$
$T' :=$ outcome of Edmonds' algorithm with input $(G', \veebar)$
Return the path from each $v \in D$ to $\veebar$ inferred by $T'$
```

Theorem 6.2 *Algorithm 2 solves* RESOLVE-DELEGATION *in a confluent manner, in polynomial time.*

Proof. It is immediate that the complexity of Algorithm 2 is polynomial in the input size. The fact that the output of Algorithm 2 is a directed temporal tree that spans the vertices of $(V \cup \{\veebar\}) \setminus A$ follows from the fact that Edmonds' algorithm returns a tree that spans the set of vertices of G', which equals $(V \cup \{\veebar\}) \setminus A$. Additionally, the result of the run of Edmonds' algorithm is a graph in which every vertex of G' is reachable from $\veebar$. Given that every edge of G' has a corresponding edge in G of reverse orientation, it holds that the output of Algorithm 2 contains a path to $\veebar$ from every vertex of $V \setminus A$. Note also that Algorithm 2 is a valid delegation rule since it assures that abstaining voters will not be asked to vote or to delegate their rights, that delegating voters will not be asked to delegate to an abstaining one and that casting voters will be asked to cast a ballot. The proof of these claims is straightforward.

[Edm67] Edmonds (1967): Optimum Branchings.

[7] To be noted that in [BDG+22], a confluent delegation rule, referred to as MinSum, has been proposed, under a more restricted voting framework compared to ours, and its polynomial time computability has been very recently established [CGN22], using an approach that is also based on Edmonds' algorithm.

We finally need to argue about the optimality of the algorithm concerning the maximization of electorate's satisfaction. This follows by the fact that any solution by Edmonds for $(G', \underline{\vee})$ corresponds to a feasible solution of RESOLVE-DELEGATION, with the same cost, and vice versa. □

We now shift our focus to efficient utility maximizing delegation rules that satisfy time-consciousness but are not necessarily confluent. Despite not necessarily resulting in a tree structure, such a rule should still suggest a precise path to a casting voter for every delegating voter v. Then, the utility of v will be derived from her immediate representative (i.e., the weight of the first edge) in that path, regardless of whether other paths going through v may exist for serving other voters who have delegated to v. The question of why non-confluent delegation rules merit investigation is discussed in [BDG+22]. It was discovered that, among a large family of delegation rules, only non-confluent rules possessed the potential to satisfy the axiom of *copy-robustness*, an axiom that is also motivated by practical considerations [BS15], which, at a conceptual level, guarantees that neither a delegating voter nor her representative can be better off if the former casts an identical ballot to the latter instead of delegating. Moreover, there are non-confluent rules with desirable properties that have been previously studied, such as the Depth-First-Delegation rule that precludes the possibility of Pareto-dominated delegations [KR20]. Hence, it is not unprecedented to sacrifice confluence on the altar of attaining other desirable attributes. However, quite surprisingly, even in the absence of a requirement for a confluent rule, RESOLVE-DELEGATION remains NP-hard, as shown by the following theorem. Notably, the result holds even for simple scenarios that involve only a brief deliberation phase, uniform trust-horizon parameters across all voters and a lone delegating voter. It is orthogonal to the result of Theorem 6.1, since it explicitly uses the fact that the considered elections are not of retrospective trust.

[BDG+22] Brill et al. (2022): Liquid Democracy with Ranked Delegations.

[BS15] Behrens and Swierczek (2015): Preferential Delegation and the Problem of Negative Voting Weight.

[KR20] Kotsialou and Riley (2020): Incentivising Participation in Liquid Democracy with Breadth-First Delegation.

Theorem 6.3 *RESOLVE-DELEGATION in a time-conscious manner is NP-hard, even for profiles with only a single delegating voter.*

Proof. Given a graph $G(V \cup \{\underline{\vee}\}, E, \tau, L, w, \delta)$ that models the deliberation phase of a t-LD election and a parameter k, we call Π the decision variant of the problem RESOLVE-DELEGATION in a time-conscious manner, which asks for the existence of a solution with total satisfaction at least k. At what follows, we provide a reduction to Π, from a problem called RESTLESS TEMPORAL PATH, that was shown to be NP-hard[8] in [CHM+21].

In the RESTLESS TEMPORAL PATH we are given, a temporal graph $G'(V', E', \tau', L')$, two distinct vertices x, y of V' and an integer Δ. The question is to determine whether a Δ-restless path from x to y exists in G', where a temporal path $p = (v'_{i-1}, (e'_i, t_i), v'_i)_{i \in [\ell]}$ is called Δ-restless if for all $i \in [\ell]$, it holds that $e'_i = (v'_{i-1}, v'_i)$, $t_i \in \tau'(e'_i)$, and for all $i \in [\ell - 1]$ we have $t_i \leq t_{i+1} \leq$

[8] The result from [CHM+21] pertains to the undirected variant of the problem. Nevertheless, an analogous result can be shown for the directed variant by a straightforward transformation.

[CHM+21] Casteigts et al. (2021): Finding Temporal Paths Under Waiting Time Constraints.

$t_i + \Delta$ (in case $t_1 + \Delta > L$, the rightmost inequality is trivially satisfied); an example of such a path is provided in Section 6.A. The hardness holds even for temporal graphs with a lifespan of 3 time instants and even if $\Delta = 1$.

Given an instance $(G'(V', E', \tau', L'), x, y, \Delta)$ of RESTLESS TEMPORAL PATH we create an instance $(G(V \cup \{\veebar\}, E, \tau, L, w, \delta), k)$ of Π as follows:

- let $L = L' + 1$,
- let G be the reversed graph of G', i.e. $V = V'$ and for every edge $e' = (v, u)$ of E' we add an edge $e = (u, v)$ in E such that $\tau(e) = \tau'(e')$ and $w(e) = 1$,
- we add in the vertex set of G a special vertex $\veebar$ and in E an edge $(x, \veebar)$ with $w((x, \veebar)) = 0$ and $\tau((x, \veebar)) = [1, L]$,
- for every edge (v, u) and for every time-instant $t \in [L]$ such that $v \neq x$ and $t \in \tau((v, u))$, we set $\delta_v^t = \min\{t - 1, \Delta\}$,
- we add a dummy vertex a with no out-going edges and an edge (y, a) with $w((y, a)) = 1$ and $\tau((y, a)) = [L, L]$,
- we set $k = 1$.

We proceed with a few observations on the profile of t-LD elections that G models. First, we have that at time L all voters that correspond to vertices other than x, y want to abstain, since there are no edges in G' that are present at time $L' + 1$. On the other hand, the only edge towards $\veebar$ in G has x as its head, and therefore, x corresponds to the only casting voter. Therefore, the only delegating voter in the created instance is y, since it corresponds to the only vertex that has an outgoing edge to a vertex other than $\veebar$, at the final time-instant. Furthermore, since all approved representatives of y are tied in the first place of her ranking, and given that the only edge towards $\veebar$ is from x, asking for a time-conscious solution with a total utility of at least 1 is equivalent to determining whether a time-conscious path from y to x exists.

For the forward direction, say that p is a directed Δ-restless path from x to y that verifies that (G', x, y) is a YES-instance of the RESTLESS TEMPORAL PATH problem. Then, the edges of G that correspond to edges of p in G' induce a path from y to x in G, since they are of reverse orientation, and consequently a path from y to $\veebar$. To prove that the path, called p', from y to $\veebar$ is δ-time-conscious, we focus on an arbitrary pair of consecutive edges of p, namely $((u, v), t'), ((v, z), t'')$. Given that p is Δ-restless in G', it holds that $t' \leq t''$ and that $t'' \leq t' + \Delta$. But then, by the construction of G, the path p' includes the following two edges: $((z, v), t''), ((v, u), t')$, for which $t'' \geq t' \geq t'' - \Delta = t'' - \delta_z^{t''}$. By applying the same argument for each pair of consecutive edges of p, we conclude that p' is indeed δ-time-conscious.

For the reverse direction, say that $(G(V \cup \{\veebar\}, E, \tau, L, w, \delta), k)$ is a YES-instance of Π, or in other words that there exists a δ-time-conscious path from y to $\veebar$. Given that the only edge towards $\veebar$ is from x and that it is present at any time-step of the deliberation phase, there also exists a δ-time-conscious path, say p, from y to x. In analogy to the forward direction, selecting an edge (u, v) of E' if and only if the edge (v, u) of E belongs to p, establishes a Δ-restless path from x to y in G', verifying that $(G'(V', E', \tau', L'), x, y)$ is a YES-instance of RESTLESS TEMPORAL PATH. □

Continuing with our study of efficient utility-maximizing delegation rules that are time-conscious but not necessarily confluent, we now turn to exploring potential workarounds to the impossibility result of Theorem 6.3. To overcome the intractability, we restrict ourselves to the still hard variant where the voters share the same time-horizon parameter and propose the following relaxations:

(a) Assuming retrospective trust profiles, i.e. $\delta_v^t = t - 1$, for every voter v and every time-step t, in which v delegates. These profiles are motivated by the fact that in real life, as long as voters do not change their opinion arbitrarily, it is likely that a delegating voter will trust another for all the past steps instead of taking the effort to further refine her trust interval.

(b) Permitting walks instead of only paths, or in other words allowing for revisits to vertices, along a path from a delegating voter to a casting one. This enlarges the solution space and can be helpful towards achieving time-consciousness in certain instances, as it may be necessary to go through a cycle before being able to satisfy the time constraints.[9]

The approach of neglecting confluence, enables the development of local delegation rules, likewise the rules studied in [CM19], that make a decision for every voter completely independent of the choices made for the rest of the electorate. For the two relaxations suggested in the previous discussion, we suggest a simple procedure that, in high level, visits every vertex v, corresponding to a delegating voter v, in a sequential manner, and for each such vertex, it detects a feasible, i.e. δ-time-conscious, way to reach $\veebar$, that uses the out-going edge of v of maximum possible weight. The aforementioned way of reaching a casting voter can be computed by a suitable modification of the temporal analog of the Breadth-First search algorithm from [MMS19], in the case where the input profile is of retrospective trust and by using the polynomial procedure that is based on Dijkstra's algorithm, from [BHN+20], in the case where walks are allowed and all voters share the same trust-horizon parameter.

Concerning the first relaxation, in [MMS19], a polynomial-time algorithm was suggested to solve a (more general than what we need in our setting) problem, called FOREMOST PATH. In this, we are given a (unweighted) directed temporal graph $G(V, E, \tau, L)$, a source vertex $v \in V$, a sink vertex $u \in V$, and a time-

[9] For an illustration, we refer to Section 6.A.

[CM19] Caragiannis and Micha (2019): A Contribution to the Critique of Liquid Democracy.

[MMS19] Mertzios et al. (2019): Temporal Network Optimization Subject to Connectivity Constraints.

[BHN+20] Bentert et al. (2020): Efficient Computation of Optimal Temporal Walks Under Waiting-Time Constraints.

instant $t_{start} \in [L]$, and we are asked to compute[10] a time-respecting path from v to u, that starts no sooner than t_{start} (or report that such a path does not exist). Recall that, the definition of a time-respecting path has been provided in the proof of Theorem 6.1.

For the second relaxation, we begin by the following definition: Given, a temporal graph $G(V, E, \tau, L, \delta)$ in which all entries of the vector δ coincide with a fixed value Δ, a temporal walk p of G of length ℓ, say $p = (v_{i-1}, (e_i, t_i), v_i)_{i \in [\ell]}$ such that v_i's are not necessarily all pairwise distinct, is called Δ-restless if for every $i \in [\ell]$ it holds that $e_i = (v_{i-1}, v_i)$ and that $t_i \in \tau(e_i)$ and for $i \in [\ell - 1]$ it holds that $t_i \leq t_{i+1} \leq t_i + \Delta$. To solve efficiently the relaxation of RESOLVE-DELEGATION in a time-conscious manner, when walks are allowed, we utilize the procedure from [BHN+20], that outputs[11] a Δ-restless temporal walk between two vertices, for any fixed Δ.

For compactness, we provide a unified presentation of the positive results, under Algorithm 3, which handles both of the above relaxations. In the statement of this algorithm, we will use the term *journey* to refer either to a path when dealing with the first relaxation or to a walk when discussing the second one.

Algorithm 3: A time-conscious and efficient utility maximizing delegation rule for input $G(V \cup \{\underline{\vee}\}, E, \tau, L, w, \delta)$ applicable for profiles of retrospective trust or profiles in which walks are allowed.

```
For every edge $e \in E$:
  replace $\tau(e)$ with $\{L + 1 - t : t \in \tau(e)\}$
For every vertex $v \in V \setminus A$:
  $E_v := \{(v, u) \in E : u \in (V \cup \{\underline{\vee}\})\}$
  While a journey from $v$ to $\underline{\vee}$ hasn't been found and $|E_v| > 0$:
    If $(v, \underline{\vee}) \in E_v$: pick $(v, \underline{\vee})$ as the path from $v$ to $\underline{\vee}$, and exit
    the while loop
    $\tilde{e} := \texttt{argmax}\{w(e) : e \in E_v\}$
    If walks are allowed:
      $G' := (V \cup \{\underline{\vee}\}, (E \setminus E_v) \cup \{\tilde{e}\}, \tau, L, \delta)$
      search for a $\Delta$-restless walk from $v$ to $\underline{\vee}$ in $G'$, if it
      doesn't exist, remove $\tilde{e}$ from further consideration
    Else:
      $G' := (V \cup \{\underline{\vee}\}, (E \setminus E_v) \cup \{\tilde{e}\}, \tau, L)$
      $t_{start} := s_{\tilde{e}}$, where $\tau(\tilde{e}) = [s_{\tilde{e}}, t_{\tilde{e}}]$
      solve FOREMOST PATH for $(G', v, \underline{\vee}, t_{start})$, if a solution
      is not found, remove $\tilde{e}$ from further consideration
Return the set of determined journeys
```

Theorem 6.4 *Algorithm 3 solves* RESOLVE-DELEGATION *in a time-conscious manner, in polynomial time, for profiles of retrospective trust. Moreover, the same holds for the variant where walks are allowed, for profiles in which there is a common, fixed trust-horizon parameter, for all voters and time-steps.*

[10] To be more precise, the goal is to select the path that minimizes the arrival time but for our purposes, this objective is superfluous (but harmless).

[11] Once again, the problem studied in [BHN+20] is more general than the problem we need to consider here, both in terms of the input graph and the optimization objective(s).

Proof. It is immediate that the complexity of Algorithm 3 is polynomial in the input size and that the procedure returns a valid delegation rule as it does not ask abstaining voters to vote or delegate their rights, it does not ask a delegating voter to delegate to an abstainer, and it asks every casting voter to cast a ballot. Therefore, we only need to prove that the output is indeed a set of feasible δ-time-conscious paths from all delegating voters to $\underline{\vee}$, that maximizes the electorate's utility.

We begin by proving that the journeys returned by Algorithm 3 indeed maximize the electorate's utility, assuming that the if and else blocks in lines 8 and 11 output a feasible solution, whenever there exists one. To that end, we first note that since it is sufficient to return a journey from a vertex v to $\underline{\vee}$ that is totally independent from the returned journey from every other vertex to $\underline{\vee}$, the maximization of the utility of the electorate boils down to the problem of maximizing the utility of every individual voter. Consider now a delegating voter v. Since Algorithm 3 examines the out-going edges of v in order of decreasing weight, and it stops at the very first time it finds a feasible solution for v, this is obviously a solution of maximum utility for v.

To prove feasibility of the outcome, starting from the case where the input profiles are of retrospective trust, we claim that the replacement of the edge time-labels ensures that a time-respecting path from v to $\underline{\vee}$ in G' corresponds to a δ-time-conscious path from v to $\underline{\vee}$ in G. Say that two consecutive edges e_1' and e_2' are being used in such a time-respecting path, at time-instants t_1' and t_2' respectively. Then, $t_1' \leq t_2'$. Equivalently, there are two consecutive edges in G, say e_1 and e_2 that are present at time-instants t_1 and t_2 respectively. Since $t_1' \leq t_2'$, it also holds that $L + 1 - t_1 \leq L + 1 - t_2$ and therefore, $t_1 \geq t_2$. Hence e_1 can be placed before e_2 in a δ-time-conscious path in G, provided that the input is of retrospective trust. By the optimality of the outcome of the algorithm for solving FOREMOST PATH, from [MMS19], one can deduce a feasible time-respecting path from v to $\underline{\vee}$ in G', which in turn implies the existence of a δ-time-conscious path from v to $\underline{\vee}$ in G.

Coming now to the case where walks are allowed, the feasibility follows by the algorithm suggested in [BHN+20] for determining whether a Δ-restless walk between two specified vertices exist. We need again to use similar arguments to the ones used for the retrospective trust case, showing that the replacement of the edge time-labels in the first lines of Algorithm 3, ensures that a Δ-restless path from v to $\underline{\vee}$ in G' corresponds to a δ-time-conscious path from v to $\underline{\vee}$ in G. □

Finally, we conclude with studying RESOLVE-DELEGATION in a time-conscious and confluent manner, but now without the requirement of computational efficiency. Clearly, if polynomial solvability is no longer a worry, a straightforward brute-force procedure, which is exponential in the number of edges (and hence in n)

and also in L, can examine all possible trees to maximize the voters' satisfaction. However, our objective goes beyond this. We aim at developing a procedure that could be well-suited for scenarios where the deliberation phase is prolonged, avoiding exponentiality in L. We focus on designing an algorithm with a running time exponentially dependent only on the number of delegating voters $|D|$ (upper bounded by n), which would still be suitable in any relatively small community. Yet, this is not possible without further assumptions, given the negative result of Theorem 6.3, that holds even for a single delegating voter. Therefore, as before, we resort to instances of t-LD elections of retrospective trust.

Theorem 6.3: RESOLVE-DELEGATION in a time-conscious manner is NP-hard, even for profiles with only a single delegating voter.

Theorem 6.5 *RESOLVE-DELEGATION in a time-conscious and confluent manner is solvable in time exponential in $|D|$ and polynomial in the remaining input parameters, for profiles of retrospective trust.*

Proof. We suggest a procedure which consists of two components: a reduction from RESOLVE-DELEGATION in a time-conscious and confluent manner to the DIRECTED MINIMUM STEINER TREE (D-MST) problem and an execution of an appropriate algorithm for the latter.

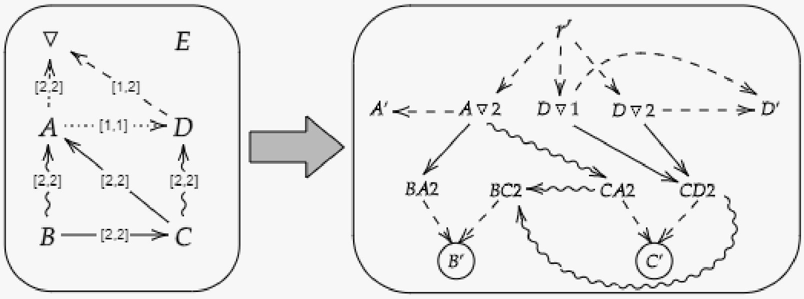
Given a graph $G(V \cup \{\underline{\vee}\}, E, \tau, L, w, \delta)$ that models the deliberation phase of a t-LD election of retrospective trust, we refer to Π as the RESOLVE-DELEGATION problem in a time-conscious and confluent manner. We will present a reduction from Π to D-MST. Consider an instance of Π, say $G(V \cup \{\underline{\vee}\}, E, \tau, L, w, \delta)$. We will now construct an instance $(G'(V', E', w'), r', \hat{V})$ of D-MST; an example of the proposed construction will follow.

In D-MST, we are given a (static) directed edge-weighted graph $G'(V', E', w')$, where $w' : E' \to \mathbb{N}$, a source $r' \in V'$, a set of vertices $\hat{V} \subseteq V'$ called terminals, and we are asked for a subgraph of G' that includes a directed path from r' to any (terminal) vertex of $\hat{V}$, of minimum possible total weight.

- we add a source r' in V', that corresponds to $\underline{\vee}$,
- we add a (terminal) vertex v' in $\hat{V}$, for every vertex v of D, a (non-terminal) vertex v' in $V' \setminus \hat{V}$, for every vertex v of C and we call all such vertices "special",
- for every edge $e = (u, v) \in E$ such that $u \in D$ and $v \in (V \cup \{\underline{\vee}\}) \setminus A$ and for every $t \in \tau(e)$, we add a (non-terminal) vertex named (e, t) in $V' \setminus \hat{V}$,
- for every pair of edges $e_1 = (u, v), e_2 = (v, z)$ of E, such that $u, v, z \in (V \cup \{\underline{\vee}\}) \setminus A$ and for every $t_1 \in \tau(e_1)$ and $t_2 \in \tau(e_2)$ with $t_1 \geq t_2$, we add in E' a directed edge from (e_2, t_2) to (e_1, t_1) (provided that these vertices exist) of weight $max(u) - w(e_1) + min(u)$, where $max(u)$ (resp. $min(u)$) is the maximum (resp. minimum) weight of out-going edges of u, available at any time-instant,
- for every edge $e = (v, \underline{\vee}) \in E$ and for every $t \in \tau(e)$, we add in E' a directed edge from r' to (e, t) of zero weight,

➛ for every edge $e = (u, v) \in E$ such that $u \in D$ and $v \in (V \cup \{\underline{\vee}\}) \setminus A$ and for every $t \in \tau(e)$, we add in E' a directed edge from (e, t) to u' of zero weight.

In the following figure, it is illustrated the weighted directed temporal graph of an instance of problem Π (left) that corresponds to a t-LD election with 1 abstaining voter (E), 2 casting voters (A, D) and 2 delegating voters (B, C) and the weighted directed static graph of the corresponding instance of D-MST (right), where circled vertices indicate terminals. Dashed, curly and straight edges are of weights $0, 1$ and 2, respectively.



At what follows we will prove that every feasible solution for an instance of Π, given by a graph G of total utility at least k, implies the existence of a feasible solution for D-MST in $(G', r', \widehat{V})$ of cost at most k', and vice versa, where $k' = \sum_{u \in D}(max(u) + min(u)) - k$. We start with the forward direction. We need to show that given a δ-time-conscious temporal tree of G, called T, which spans the vertices of $V \setminus A$, is rooted at $\underline{\vee}$ and its total weight is at least k, one can deduce a feasible solution T' of weight at most k' in the created instance of D-MST. Obviously T induces a path from every vertex $v \in V \setminus A$ to $\underline{\vee}$, which is of unit length if $v \in C$, and of length greater than 1, otherwise. For such a path of unit length, there is also a directed path from r' to v' in G', that is of zero cost, which is formed by the edges $(r', ((v', \underline{\vee}),\mathrm{L}))$ and $(((v', \underline{\vee}),\mathrm{L}), v')$. Hence, the non-terminal vertex that corresponds to a casting voter, can be reached from r' with no cost, using exactly 2 edges. Let all such pairs of edges belong to T'.

We move on to temporal paths of length greater than 1 in T. We select an arbitrary path p of T having $\underline{\vee}$ as its sink vertex and a vertex u_0, that corresponds to a delegating voter, as its source, and we rename its vertices and edges so as $u_q = \underline{\vee}$ and $p = (u_{i-1}, (e_i, t_i), u_i)_{i \in [q]}$, for some $q \in [n]$. Since p is a δ-time-conscious path, for $i \in [q]$, it holds that $e_i = (u_{i-1}, u_i)$, $t_i \in \tau(e_i)$ and, furthermore, for $i \in [q-1]$ it holds that $t_i \geq t_{i+1}$. By the construction, there also exists a path p' in G' that can be expressed by the sequence of vertices

$$\{r', ((u'_{q-1}, u'_q), t_q), ((u'_{q-2}, u'_{q-1}), t_{q-1}), ..., ((u'_1, u'_2), t_2), ((u'_0, u'_1), t_1), u'_0\}.$$

We have constructed a directed path from r' to u_0', and u_0' is a terminal vertex since u_0 corresponds to a delegating voter. By repeating the same procedure for every path p of T that has $\veebar$ as its source, we can form a subgraph of G', say T', that consists of paths from r' to every terminal vertex.

It remains to be proven that T' is a tree of cost no more than k'. Suppose that the undirected variant of T' contains a cycle. By a similar reasoning to the above, we can prove that the undirected variant of T would also contain a cycle, which is a contradiction. We will now prove that if the total weight of the edges in T is at least k then the weight of the edges in T' is at most k'. Observe that for every vertex $v \in V \setminus A$, there is only one other vertex $u \in (V \cup \{\veebar\}) \setminus A$, such that (v, u) in T (and similarly for T') and say that, for convenience, the weight of an edge (v, u) that belongs to T is denoted by $w(v) = w(v, u)$. The total weight of the edges in T is $\sum_{v \in D} w(v) \geq k$. We focus on a pair of paths of T and T', namely p and p' respectively. Say that $p = (u_{i-1}, (e_i, t_i), u_i)_{i \in [q]}$ and that p' is formed by

$$\{((u'_{q-1}, u'_q), t_q), ((u'_{q-2}, u'_{q-1}), t_{q-1}), \dots, ((u'_1, u'_2), t_2), ((u'_0, u'_1), t_1)\}.$$

The weight of all edges in p equals $w(e_1) + w(e_2) + \dots + w(e_{q-1}) + w(e_q) = w(u_0) + w(u_1) + \dots + w(u_{q-2}) + w(u_{q-1})$. On the other hand, the weight of all edges in p' equals $max(u_{q-1}) - w(u_{q-1}) + min(u_{q-1}) + max(u_{q-2}) - w(u_{q-2}) + min(u_{q-2}) + \dots + max(u_0) - w(u_0) + min(u_0)$. Hence, for every such pair of paths, if w_p is the total weight of the path p (similarly for p') and if N_p is the set of non-sink vertices of path p, i.e. $N_p = \{u_0, u_1, \dots, u_{q-1}\}$, then $w_{p'} = \sum_{u \in N_p} (max(u) + min(u)) - w_p$.

Any rooted directed (towards the root) tree with ℓ leaves can be divided into paths $P_1, P_2, \dots, P_\ell$, such that every path has a leaf as its source and every vertex (other than the root) belongs to exactly one path as a non-sink vertex. These paths can be created with the following procedure: Initially say that only the root of the tree belongs to a set X. Select as P_1 any path from a leaf to the root and say that from now on, X also contains the vertices of P_1. For $i = 2$, select the path that does not use any vertex of X as a non-sink vertex, starting from an unexplored leaf and ending at a vertex of X and call it P_i; add the vertices of P_i to X and repeat for $i = 3, \dots, \ell$.

Therefore, T can be represented as a collection of, say $\ell(T)$, such paths in a way that each of its vertices (other than the root) appears in that collection as a non-sink vertex exactly once. If we call w_{P_i} the total weight of edges in P_i, the cost of T can be expressed as $\sum_{i \in [\ell(T)]} w_{P_i}$, and the cost of T' equals

$$\sum_{i\in[\ell(T)]}\Big(\sum_{u\in N_{P_i}}(max(u)+min(u))-w_{P_i}\Big)\leq$$
$$\sum_{u\in D}(max(u)+min(u))-k=k'.$$

For the reverse direction, we firstly note that it is without loss of generality to assume that any optimal D-MST of G' contains a path from $\veebar$ to any special (terminal or non-terminal) vertex, since every special non-terminal vertex v' can be reached from r' at no cost by following the edges $(r',((v',\veebar),L))$ and $(((v',\veebar),L),v')$, which definitely exist. The rest of the proof follows from the same arguments presented in the forward direction.

Given an instance $(G'(V',E',w'),r',\hat{V},k')$, the D-MST can be solved in time $O^*(3^{|\hat{V}|})$ by a modification of the classic algorithm for the (undirected) MINIMUM STEINER TREE problem [DW71], where the O^* notation denotes the suppression of factors polynomial in the input size. Further improvements on the running time have also been suggested, as outlined in greater detail in [JLR+17]. By our construction it holds that $|\hat{V}|=|D|$. Hence, RESOLVE-DELEGATION in a time-conscious and confluent manner is solvable in time exponential only in the number of delegating voters of the instance, for t-LD elections of retrospective trust. □

[DW71] Dreyfus and Wagner (1971): The Steiner Problem in Graphs.

In summary, within this section, we proved that while fulfilling all the desired attributes concurrently seems implausible, positive results can indeed be attained in various natural directions.

6.3 Concluding Discussion and Future Directions

Briefly speaking, the main features of Liquid Democracy, as argued in [BZ16], are the (i) voters' ability to cast a ballot, (ii) ability to delegate voting rights, (iii) transitivity of delegations, (iv) ability to modify or recall a delegation. Our work is the first in the literature, that studies a model satisfying every each of the above by suggesting delegation rules designed to prevent ballot loss. Inspired by the proposal of [BKN+22] to add a temporal dimension in the algorithmic considerations of LD models, and building upon [BDG+22], we studied a LD framework from a viewpoint that lies in the middle ground between algorithmic and axiomatic approaches. We intentionally gave significant emphasis on developing a general model for incorporating temporal aspects and we feel it opens up the way for several promising avenues for future research. One such direction is to examine whether time-consciousness (or other time-related axioms) is compatible with other axioms within LD frameworks, beyond confluence. Also an intriguing topic is to extend our positive results to further realistic families

[BZ16] Blum and Zuber (2016): Liquid Democracy: Potentials, Problems, and Perspectives.

[BKN+22] Behrens et al. (2022): The Temporal Dimension in the Analysis of Liquid Democracy Delegation Graphs.

[BDG+22] Brill et al. (2022): Liquid Democracy with Ranked Delegations.

of instances or to further generalizations of t-LD elections, e.g., when the voters are able to use a more powerful language to express complex preferences. Furthermore, studying an egalitarian objective instead of the social welfare or imposing restrictions on the maximum in-degree or on the maximum path-length in the output of a delegation rule, are perfect candidates for future works on this area. On top of this, it is also interesting to incorporate the preferences of the delegating voters over their "final" casting voter representative, instead of their immediate representative in the output graph. Finally, we also identify as an important future work topic the use of experimental or empirical evaluations of LD frameworks that take into account temporal considerations.

6.A Appendix: Illustrative Examples

This section aims to clarify some of the technical terms and concepts defined and used in Sections 6.1 and 6.2, through simple illustrative examples. The graphs in Figure 6.1 will be used as references throughout this section. For ease of presentation, we have labeled each edge of the graphs in Figure 6.1 with the corresponding time instant in which the edge is available, since in these particular examples, we have assumed that each edge is available for only one time instant. Specifically, we use the label t to denote the time interval $[t, t]$.

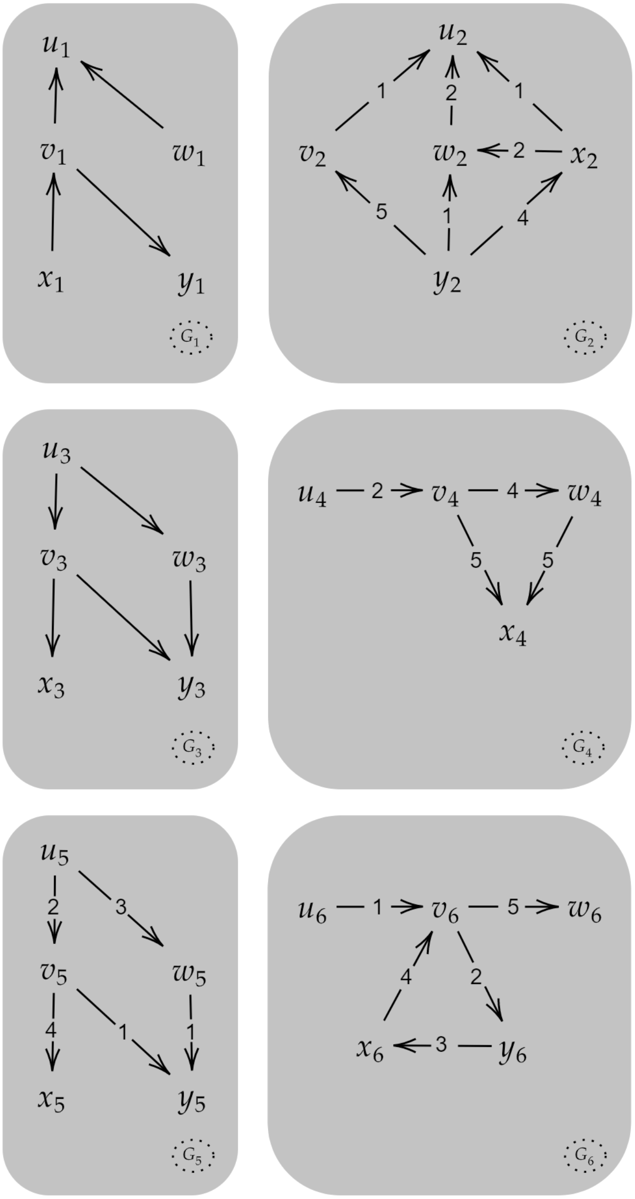


Figure 6.1: Graphs that serve to demonstrate key concepts and terminology used in our work. In the order presented, they correspond to the following concepts: rooted directed trees (G_1), time-conscious paths (G_2), confluent delegation rules (G_3), restless paths (G_4), time-respecting paths (G_5), and restless walks (G_6). An edge labeled t is available only during the time-interval $[t, t]$.

- **Rooted directed trees.** In G_1, we observe a static variant of a temporal directed tree that cannot be said to be rooted at any of its vertices. However, by replacing the edge (v_1, y_1), with its opposite direction, a directed tree rooted at u_1 is being produced.

- **Time-conscious paths.** Consider G_2 to be a temporal graph with a lifespan of 5, and say that $d_v^t = 2$ for every vertex v and for every time-instant $t \in 1, 2, \ldots, 5$. In this case, the only δ-time-conscious path from y_2 to u_2 is (y_2, x_2, w_2, u_2). On the other hand, if $d_{y_2}^t = 4$ for every time-instant $t \in 1, 2, \ldots, 5$, then all paths from y_2 to u_2 except (y_2, w_2, u_2) are δ-time-conscious.

- **Confluent vs non-confluent rules.** Suppose G_3 represents the unweighted variant of the static graph that arises from a temporal graph that models a t-LD election in which x_3 and y_3 are casting voters, w_3 is an abstaining voter, and x_3 is the most preferred representative of v_3. Any confluent voting rule should output either (v_3, x_3) or (v_3, y_3), but not both, and should not output (w_3, y_3). However, if the path that is formed by the edges (u_3, v_3) and (v_3, x_3) is an infeasible option (perhaps due to the time-horizon parameters of the voters), a non-confluent voting rule might propose x_3 as the representative for v_3 and y_3 as the representative for u_3, via any available path.

- **Restless paths.** The notion of Δ-restless paths has been used in the proof of Theorem 6.3. In G_4, there are two possible paths from u_4 to x_4, but only the path that goes through w_4 is 2-restless.

- **Time-respecting paths.** This concept has been crucially utilized in the proof of Theorem 6.1, as well as in Algorithm 3. In G_5, there are three paths of length 2, but only the one that uses the edges (u_5, v_5) and (v_5, x_5) is time-respecting.

- **Restless walks vs paths.** The importance of allowing walks instead of paths in the search for restless journeys becomes evident when examining G_6. Specifically, the absence of an 1-restless path from u_6 to w_6 contrasts with the existence of an 1-restless walk, by utilizing the cycle between v_6, x_6, y_6, highlighting the potential for creating more feasible options with the latter approach.

7 On the Potential and Limitations of Proxy Voting: Delegation with Incomplete Votes

Broadly speaking, an election refers to a voting system in which a set of participants express their preferences over a set of possible issues or outcomes, and those are aggregated into a collective decision, typically with a socially desirable objective in mind. Besides their "traditional" applications such as parliamentary elections or referenda, elections often underpin the livelihood of modern systems such as blockchain governance [CS21; KL22] or participatory budgeting [AS21; Cab04]. Quite often, voters are called to vote on an extremely high number of issues, rendering the accurate expression of their preferences extremely challenging. For instance, the Cardano blockchain uses Project Catalyst (https://projectcatalyst.io) to allocate treasury funds to community projects, and routinely receives several thousands of proposals per funding round. Another application comes from platforms of civic participation, where the users express support on opinions or proposals [HKP+23].

An unfortunate consequence of these election scenarios is that the voters will inevitably have a confident opinion only for a small number of issues (henceforth *proposals*), as investing enough time and effort to inform themselves about the sheer volume of proposals is clearly prohibitive. In turn, the "direct voting" outcome, even under the best intentions, will most likely be ineffective in capturing the desires of society, which it would, had the voters been sufficiently informed. A well-documented possible remedy to this situation is to allow for *proxy voting* [CMM+17], a system in which the voters *delegate* their votes to *proxies*. The idea is that those proxies have the time and resources to study the different proposals carefully, and vote on behalf of the voters they represent. This in fact captures voting applications more broadly, where the reason for delegation might be a reluctance to express an opinion, lack of specialized knowledge, or even limited interest. When those proxies are part of an electorate together with other voters and proxies, the resulting system is known as *Liquid Democracy* (see also Chapters 5 and 6). Liquid Democracy has been scrutinized, with arguments presented in its favor [BDD+21; HHJ+23; RHB+22] and against it [CM19; KMP21], and at the same time it is being employed in real-world situations [Pau20] including settings similar to the one studied here, like Catalyst.

A takeaway message from the ongoing debate around delegative voting is that such processes might indeed be useful under the right circumstances. Extending this line of thought and motivated by the scenarios presented above, we aim to identify the *potential* and *limitations* of proxy voting with regard to achieving socially desirable outcomes, in settings with incomplete votes. More precisely, in this chapter we aim to characterize what is theoretically possible with delegation, and what is impossible, even under idealized conditions.

Our Setting

We focus on elections in which the aim is to choose one proposal to be implemented from a range of multiple proposals. We introduce a model of proxy voting, where voters have intrinsic approval preferences over all proposals, which are only partially revealed or known to the voters themselves. A set of *delegation representatives (dReps)* can then advertise ballots over the proposals and the voters in turn may either delegate to a proxy, if there is sufficient agreement (i.e., over a certain *agreement threshold* between the proxy's advertised ballot and the voter's revealed preferences), or vote directly. The outcome of the election is the proposal with the largest approval score, assembled by the score from the ballots of the dReps (representing voters who delegated their vote) and the voters that vote directly. The core question we pose follows:

> *"Is it possible for the dReps to advertise their preferences appropriately such that the outcome of the election has an approval score that is a good approximation of the best possible approval score; which would only be achievable if all voters had full knowledge of their preferences?"*

"Best-Case Scenario"

We study the aforementioned question by making the following assumptions:

1. The dReps are fully informed about the preferences of the voters, i.e., they know exactly the vector of intrinsic preferences for each voter.
2. The dReps themselves do not have actual preferences and their only goal is to achieve the best possible approximation of the optimal approval score. To do so they coordinate with each other and advertise their types accordingly.
3. When there are multiple proposals with the maximum revealed score, ties are broken in favor of those with the highest intrinsic scores.

One should not of course expect all of these assumptions to apply in practice. We would expect the dReps to be only partially informed about the preferences of the voters (e.g., via some probabilistic model) and to exhibit some sort of rational behavior (e.g., needing to be appropriately incentivized to advertise ballots that are aligned with socially-desirable outcomes). Still, studying "best-case scenario" is already instructive for results in all other regimes. In particular:

- ➛ Our *negative results (inapproximability bounds)* immediately carry over to other settings as well, regardless of the choice of the dRep information model, the rationality model for the dReps, or the choice of the tie-breaking rule. In other words, we show that certain objectives are *impossible*, even when a set of fully-informed dReps coordinate to achieve the best outcome, hence they are certainly impossible for any other meaningful setting.

➛ Our *positive results (approximation guarantees)* establish the limits of the aforementioned impossibilities: if something is not deemed impossible by our bounds, it should be the starting point of investigations for an information/rationality model chosen for the application at hand. Clearly, if our upper bounds establish that a certain number of dReps suffices to achieve a certain approximation in the "best-case scenario" setting, one should expect a slightly larger number of dReps to be needed in practice.

Contribution.

We firstly present a strong impossibility, namely that for any agreement threshold higher than 50%, the best achievable approximation ratio is linear in the number of voters. On the positive side, we show that for an appropriate *coherence* notion of the instance, capturing the commonalities of the set of proposals that sets of voters are informed about, meaningful approximations are possible. For the natural case of an agreement threshold of 50%, we show that a single dRep is capable of achieving an approximation factor of 3, whereas only 2 dReps are sufficient to elect the optimal proposal. Most significantly, we present general theoretical upper and lower bounds on the achievable approximation guarantees, depending on the agreement threshold, the number of dReps, and the coherence of the instance.

Related Work. We begin by commenting on some works that are closer in spirit to ours. Meir et al. [MST21] propose a model with a similar objective, focusing on the analysis of *sortition*, i.e., the approximation of the welfare achieved by selecting a random small-size committee. In a related direction, Cohensius et al. [CMM+17] analyze particular delegation mechanisms, under elections with samples of voters located randomly in a metric space, according to some distribution. Our approach does not consider any randomization, neither for the voting rule nor for the preferences. Finally, Pivato and Soh [PS20b] also consider the performance of proxy voting, focusing on understanding when the proxy-elected outcome coincides with the outcome of direct voting. Again, the model in [PS20b] is randomized, where the voters delegate based on the probability of agreement to a proxy, and not based on a deterministic distance function. Moreover, no analysis of approximation guarantees is undertaken in [PS20b]. Our work can be seen as one that contributes to the corpus of findings in favor of proxy voting frameworks [BDD+21; HHJ+23; RHB+22], albeit in a markedly different manner.

There is significant work within the field of Computational Social Choice on elections with incomplete votes. One stream has focused on the identification of possible and necessary winners by exploring potential completions of incomplete profiles; see [Lan20] for an overview. Recent work has concentrated on the computational complexity of winner determination under various voting rules

[MST21] Meir et al. (2021): Representative Committees of Peers.

[CMM+17] Cohensius et al. (2017): Proxy Voting for Better Outcomes.

[PS20b] Pivato and Soh (2020): Weighted Representative Democracy.

[BDD+21] Becker et al. (2021): Unveiling the Truth in Liquid Democracy with Misinformed Voters.

[HHJ+23] Halpern et al. (2023): In Defense of Liquid Democracy.

[RHB+22] Revel et al. (2022): Liquid Democracy in Practice: An Empirical Analysis of its Epistemic Performance.

within the framework of incomplete information [BD15; IIB+22; ZYG19]. Another direction has studied the complexity of centralized interventions to reduce uncertainty [ABE+22] (e.g., by educating a selected set of voters or computing delegations via a centralized algorithm). Furthermore, there has been an exploration of the effect of minimizing the amount of information communicated [AAL+19; KKK+11] as well as of the interplay between voters' limited energy and social welfare [Ter23]. Considerable attention has been devoted to the exploration of efficient extensions of incomplete profiles to complete ones that satisfy desirable properties [EFL+15; Lac14; TKO21]. A conceptually related area focuses on distortion, investigating the implications of applying rules to preferences that are less refined than voters' intrinsic preferences [AFS+21].

7.1 Election Framework and Definitions

In the current section we formally describe the main attributes of the election setting we study.

Proposals and Voters. Let $C = \{1, 2, \dots, m\}$ be a set of candidate proposals, where for each proposal there are exactly two options: to be elected or not. Let also $N = \{1, 2, \dots, n\}$ be a set of voters responsible for determining the elected proposal. Each voter $i \in N$ is associated with *approval preferences* $v_i \in \{0, 1\}^m$ over the set of proposals; we refer those as true or *intrinsic* preferences. Here, 1 and 0 are interpreted as "accept" (or "support") and "reject" (or "oppose") a proposal, respectively.

Crucial to our model is the fact that voters do not actually know their entire intrinsic preference vector, but only a subset of it; this could be due to the fact that they have put additional effort into researching only certain proposals to verify if they indeed support them or not, but not necessarily all of them.

Formally, we will say that each voter i has *revealed* preferences $\hat{v}_i \in \{0, 1, \perp\}^m$, where $\perp$ denotes that the voter does not have an opinion on the corresponding proposal. As such, we have the following relations between v_i and $\hat{v}_i$:

$$\forall j \in C \ : \ (\hat{v}_i(j) = v_i(j)) \vee (\hat{v}_i(j) = \perp)$$

The collection of proposals for which a voter i has developed an opinion is referred to as their *revealed set*, denoted by $R_i := \{j \in C : \hat{v}_i(j) = v_i(j)\}$. Let $m_i := |R_i|$. Each voter i also has an integer weight w_i.

It is without loss of generality to assume that $w_i = 1$ for all $i \in N$, as we can simply make w_i copies of voter i (with the same preferences), and all of our results go through verbatim.

dReps. A *delegation representative (dRep)* is a "special" voter whose aim is to attract as many voters as possible to delegate their votes to her, and then participates in the election with the combined weight of those voters. In contrast with some of the literature, and consistently with the "best-case scenario" motivation (see page 135), we view dReps as unweighted agents, devoid of personal preferences over the proposals, with the responsibility of facilitating the election of a proposal that attains substantial support from the voters.

For any proposal $j \in C$, every dRep has an *advertised*, "intended" vote (or type), $t(j) \in \{0, 1\}$, which is visible by the voters. We assume here that dReps present votes for all proposals.[1] We will sometimes abuse the notation and refer by t both to the type vector of a dRep as well as to the dRep itself. The distance between a voter i and a dRep of type t is calculated using the Hamming distance function and is dependent on the revealed preferences of i and the advertised type of t in proposals that are revealed to i. Formally, let $t_{|i}$ be the projection of the type t to the proposals that belong to R_i. Then we define the distance between a voter i and a dRep t as $d(i, t) := H(\hat{v}_i, t_{|i})$.

Agreement Threshold. For a voter to delegate their vote, they have to agree with the dRep in a certain number of proposals. This is captured by a threshold bound, any agreement above which results in delegation. To make this formal, we will assume that voter i delegates their vote to a dRep when their distance to the dRep's type, taking into account only the voter's revealed preferences, is at most $\lfloor \frac{m_i - k_i}{2} \rfloor$, where k_i is a parameter quantifying the reluctance of voter i to entrust their voting power to a proxy. Obviously $k_i \leq m_i$, for every i, and we will mainly focus on scenarios in which all voters have the same parameter, thus $k = k_i$, for every voter i. For example, when $k = 0$, a voter delegates their vote if they agree with the dRep in at least half of the proposals in their revealed set; we will refer to this case as *majority agreement* (see also [CW23; FW22b] for a use of a very similar threshold in a difference context). Given a dRep of type t, we say that t *attracts* a set of voters $A(t) := \{i \in N : d(i, t) \leq \lfloor \frac{m_i - k_i}{2} \rfloor\}$. Additionally we define $A(D)$, for a set of dReps D as

$$A(D) := \{i \in N : \exists\, t \in D \text{ s.t. } d(i, t) \leq \lfloor \frac{m_i - k_i}{2} \rfloor\}.$$

If a voter is attracted by multiple dReps, we assume they delegate to any of them arbitrarily; this choice makes our positive results stronger, whereas, notably, our negative results work for any choice (e.g., even for the more intuitive choice of the closest, in terms of Hamming distance, accepted dRep).

Preference Profiles. Let $V = (v_i)_{i \in N}$ and $\hat{V} = (\hat{v}_i)_{i \in N}$. We call *intrinsic preference profile* $P = (N, C, V)$ a voting profile that contains the intrinsic preferences of the voters in N on proposals from C. Similarly, we call *revealed preference profile* $\hat{P} = (N, C, \hat{V})$ the voting profile that contains their revealed preferences. Finally, $\hat{P}_D = (N, C, \hat{V} \cup D)$ refers to the preference profile on proposals from C, that contains the revealed preferences of the voters in N as well as the advertised types of the dReps in D.

Approval Voting Winners. The winner of the election is the proposal with the highest (weighted) approval score. Formally, let $sc(j)$ denote the score of a proposal $j \in C$ in the profile P, i.e., the total weight of the voters $i \in N$ such that $v_i(j) = 1$. A proposal $j \in C$ is the winning proposal in the profile P if $sc(j) \geq sc(j'), \forall j' \in C$. Similarly, we define $\hat{sc}(j)$ and $\hat{sc}_D(j)$ to be

Tie-breaking for the winner: We assume that $\arg\max\{\cdot\}$ returns a single winning proposal rather than a winning set, according to some tie-breaking rule. Consistently with our discussion on the "best-case scenario" (see page 135), we assume that the tie-breaking is always in favor of the proposal with the maximum intrinsic score.

[1] We could also allow dReps to abstain in some proposals, and this would not make any difference in our setting.

the score of a proposal $j \in C$ in the profile $\widehat{P}$ and $\widehat{P}_D$, respectively. Note that $\widehat{sc}(j)$ represents the score that proposal j would attain if all voters were to vote directly and $\widehat{sc}_D(j)$ comprises the scores of the dReps (whose weight is the total weight of the voters they have attracted) and the scores of the voters that have not delegated their votes to any dRep, i.e., that are voting directly. Let $\texttt{win}(P) := \arg\max\{sc(j), j \in C\}$ and $\texttt{opt}(P) := sc(\texttt{win}(P))$. The same notions can be extended to profiles $\widehat{P}$ and $\widehat{P}_D$.

Our goal is to select a set of λ dReps that will collectively (by participating in the election and representing voters according to the submitted thresholds) ensure that a proposal of high intrinsic approval score will be elected. We refer to this problem generally as PROXY SELECTION.

PROXY SELECTION(P,$\widehat{P}$,k,λ)	
Given:	An intrinsic voting profile P and a revealed voting profile $\widehat{P}$ on a set C of m proposals and a set N of n weighted voters; the voters' true (resp. revealed) preferences V (resp. $\widehat{V}$); a parameter k, so that a voter i is attracted by a dRep with type t if $d(i,t) \leq \lfloor \frac{m_i - k}{2} \rfloor$; an upper bound λ on the number of dReps.
Output:	Specify type vectors for all dReps in D, with $\lvert D \rvert \leq \lambda$, such that $\texttt{win}(P) = \texttt{win}(\widehat{P}_D)$.

The performance of a suggested set of dReps is measured in terms of how well the intrinsic score of the winning proposal under their presence approximates the highest intrinsic approval score. Formally:

Definition 7.1 *Let $\rho \geq 1$. We say that a set of dReps D achieves a ρ-approximation if $sc(\texttt{win}(\widehat{P}_D)) \geq \frac{1}{\rho}\, sc(\texttt{win}(P))$.*

One might be inclined to believe that attracting a sufficiently large set of voters is enough to achieve an analogous approximation ratio guarantee, e.g., that attracting half of the voters would result in a 2-approximation. The following proposition establishes that this is not the case. It demonstrates that the attraction part is only one component towards solving PROXY SELECTION, and, therefore, achieving good approximations requires further insights.

Proposition 7.1 *It is possible for a single dRep to attract half of the voters without achieving a 2-approximation.*

Proof. Consider an instance with three voters and four proposals. Voters' intrinsic preferences are presented in the table below, where the revealed preferences are given in white background.

	I_1	I_2	I_3	I_4
voter 1	1	1	0	0
voter 2	1	0	1	1
voter 3	1	0	1	1

It is evident that $\texttt{win}(P) = I_1$, and $\texttt{opt}(P) = 3$. At the same time the dRep that votes in favor of all proposals attracts 2 out of the 3 voters, namely voter 2 and voter 3. This means that $\hat{sc}_D(I_1) = |A(D)| = 2$ (where, recall that D is the set of dReps, here consisting of the single aforementioned dRep), while $\hat{sc}_D(I_2) = 3$, since both the dRep and voter 1 vote in favor of I_2. However, $sc(I_2) = 1 = \frac{1}{3}\texttt{opt}(P)$, and consequently, the dRep that votes in favor of all proposals, attracting at least half of the voters, cannot yield an approximation factor better than 3. □

Before we proceed, we highlight that if the tie-breaking rule is not in favor of electing the optimal proposal, then the situation can be much worse; in fact, the approximation can be infinite as demonstrated by the following example.

Example 7.1 Consider an instance with n voters and m proposals. Voters' intrinsic preferences are presented in the table below, where the revealed preferences are given in white background.

	I_1	I_2	I_3	$\cdots$	I_m
voter 1	1	0	1	$\cdots$	1
$\cdots$	$\cdots$	$\cdots$	$\cdots$		$\cdots$
voter n	1	0	1	$\cdots$	1

It holds that $\texttt{win}(P) = I_1$, and $\texttt{opt}(P) = n$. Suppose that we select as dRep the one that advertises 1 for every proposal under consideration. Then, $A(D) = N$. However, the presence of this dRep in D implies that $\hat{sc}_D(j) = n$, for every proposal j. Breaking ties in an adversarial manner, results to the election of I_2, albeit $sc(I_2) = 0$, which leads to an infinite approximation.

We conclude the section with the definition of an important notion for our work, that of a *coherent set of voters*, i.e., sets of voters with the same revealed sets. Several of our positive results will be parameterized by properties of those sets, such as the size of the largest coherent set.

Definition 7.2 *A set of voters $N' \subseteq N$ is called coherent if $R_i = R_{i'}, \forall i, i' \in N'$. An instance of* PROXY SELECTION *is called coherent if N is coherent.*

Importantly, given an instance of PROXY SELECTION, it is computationally easy to verify if it is coherent, or to find the largest coherent set of voters.

Claim 7.2 *Given an instance of* PROXY SELECTION*, we can in polynomial time identify the largest coherent set and its size. Furthermore, we can in polynomial time decide if the instance is coherent.*

Proof. The first part of the statement obviously implies the second, which solely necessitates verifying whether the size of the largest coherent set is n. Now, given the intrinsic preferences P of an instance of PROXY SELECTION we can find the largest coherent set via the following simple algorithm. For every voter i, find a set of voters that can form a coherent set together with i, namely $\{i' \in N : R_i \equiv R_{i'}\}$. Among the sets obtained this way, select one of maximal cardinality. The described procedure runs in time $O(n^2 m)$. □

7.2 Approximations and Impossibilities

We start our investigation with the case of a single dRep ($\lambda = 1$). Our main result here is rather negative, namely that no matter how the dRep chooses their vote, the approximation ratio cannot be better than linear in the number of voters.

Theorem 7.3 *For a single dRep and any $k > 0$, the approximation ratio of* PROXY SELECTION *is $\Omega(n)$.*

Proof. Consider an instance with an odd number of $m > 3$ proposals and $n = m - 1$ voters, where $k_i > 0, \forall i \in [n]$, such that:

- For every voter $i \in [n]$, their preferences with respect to proposal m are as follows: $v_i(m) = 1$ and $\hat{v}_i(m) = \bot$.
- The remaining $m - 1$ proposals are partitioned in $\frac{m-1}{2}$ pairs, say $\{1, 2\}$, $\{3, 4\}, \dots, \{m-2, m-1\}$ and for each one of these $\frac{m-1}{2}$ pairs, say $\{j, j+1\}$, there are two distinct voters, namely j and $j + 1$, where voter j votes in favor of both proposals j and $j + 1$ whereas voter $j + 1$ votes in favor of j but against $j + 1$; for every other proposal j', it holds that $v_i(j') = 0$ and $\hat{v}_i(j') = \bot$, where $i \in \{j, j + 1\}$.

Say that P and $\hat{P}$ are the intrinsic and revealed profiles of the created instance, respectively. Clearly, $\texttt{win}(P) = m$ and $\texttt{opt}(P) = n$. We claim that a single dRep, called t, regardless of their advertised type, will contribute to electing a proposal i that satisfies $sc(i) \leq 2$, leading to an inapproximability of $\frac{n}{2}$. Towards this, first notice that t cannot attract both voters of any pair. This is easy to see, as a distance of $\max\{0, \lfloor\frac{m_i-k_i}{2}\rfloor\}$ for $m_i = 2$ and $k_i \geq 1$ means agreement on both revealed proposals. As a result, for any such pair of voters, t will either attract zero or one voter(s).

Consider first the scenario where $A(\{t\}) = \emptyset$. In this case, since $\hat{sc}_D(m) = 0 < \hat{sc}_D(j)$ for any proposal $j \neq m$, it holds that there exists a proposal j' such that $\texttt{win}(P_D) = j'$ for which $sc(j') = 2$, or in other words, the direct voting will lead to the election of an outcome that is being accepted by exactly two voters. On the other hand, if t attracts at least one voter, say i where i is odd (resp. i is even), then t must have voted in favor of at least one proposal apart from proposal m, namely for proposal i (resp. for proposal $i-1$), or else i would not fully agree with t. But then, voter $i+1$ (resp. $i-1$), who is not attracted by t, is also voting in favor of proposal i (resp. for proposal $i-1$). This results to a proposal i such that $\hat{sc}_D(i) = \hat{sc}_D(m)+1$. Therefore, the winning proposal is not m but a proposal i for which $sc(i) \leq 2$. □

Theorem 7.3 should be interpreted as a very strong impossibility result since it holds even in the "best-case scenario" (see page 135). A natural follow-up question is whether some meaningful domain restriction can circumvent this impossibility. For this, we will appeal to the notion of coherent sets of voters and we will show a bounded approximation guarantee that degrades smoothly as the size of the largest coherent set grows and as the agreement threshold decreases.

Theorem 7.4 *For a single dRep,* PROXY SELECTION *admits an approximation ratio of* $\min\left\{n, \frac{3n(k+2)}{2|S|}\right\}$*, where S is the largest coherent set of voters.*

Proof. In the current proof, as well as in other proofs of the chapter, we will denote by `dRep0` and `dRep1` the delegation representatives whose advertised types with respect to a proposal $j \in C$, are as follows:

$$\texttt{dRep0}(j) = \begin{cases} 1, \text{ if } j = \texttt{win}(P), \\ 0, \text{ otherwise.} \end{cases} \qquad \texttt{dRep1}(j) = 1, \; \forall j \in C.$$

We begin with the following statement. It is a direct consequence of the proof of Theorem 7.23, which we will present later on in the text, but will also come in handy for the present proof. To maintain the flow of our presentation and prevent redundant repetition of arguments, we opted to use this forward reference.

Claim 7.5 *Consider a coherent instance of* PROXY SELECTION, *for any set of proposals, and any set of voters, either $1/3$ of the voters agree with* `dRep0` *on at least half the proposals, or the winning proposal of direct voting receives a score that is at least $1/3$ of the votes.*

Next, we state and prove the following lemma.

Lemma 7.6 *In a coherent instance of* PROXY SELECTION, *either $sc(\texttt{win}(\hat{P})) \geq \frac{2n}{3(k+2)}$, or $|A(\texttt{dRep0})| \geq \frac{2n}{3(k+2)}$. Hence, an approximation factor of $\frac{3(k+2)}{2}$ can be achieved.*

Proof of Lemma 7.6. Consider a coherent instance, and let R be the set of proposals that are revealed to all voters. If $\texttt{win}(P) \in R$, then the optimal proposal will be elected by direct voting. Therefore, we focus on the case where $\texttt{win}(P) \notin R$. Suppose that there exists a proposal $j' \in C$ such that $\hat{sc}(j') \geq \frac{2n}{3(k+2)}$. But then the proof follows by the fact that $sc(win(\hat{P})) \geq \hat{sc}(win(\hat{P})) \geq \hat{sc}(j')$. So we can also assume that for every proposal $j \in C$, it holds that $\hat{sc}(j) < \frac{2n}{3(k+2)}$. To continue with the proof of the lemma, we will need the following claim.

Claim 7.7 *Consider a coherent instance, with R being the set of proposals commonly revealed to all voters. Suppose also that $\hat{sc}(j) < \frac{2n}{3(k+2)}$ for any $j \in R$. For any $r \in [k]$ and for any set $S_r \subseteq R$ of r proposals, let $Z(S_r, \texttt{dRep0})$ be the set of voters that totally agree with* `dRep0` *in all proposals of S_r. Then $|Z(S_r, \texttt{dRep0})| \geq \frac{(k+2-r)n}{(k+2)}$.*

Proof of Claim 7.7. We will prove the statement by induction on r. To prove it for $r = 1$, we define $N_j := |\{i \in N : \hat{v}_i(j) = 0\}|$, for any proposal $j \in C$. Then, by the fact that $\hat{sc}(j) < \frac{2n}{3(k+2)} < \frac{n}{(k+2)}$ it holds that $N_j \geq n - \frac{n}{(k+2)} \geq \frac{(k+1)n}{(k+2)}$, and the induction base follows. Say now that the statement holds for a fixed $r' < k$ and call $S_{r'} \subseteq R$ an arbitrary set of r' proposals.

We build a set $S_{r'+1}$ by adding to $S_{r'}$ an arbitrary proposal $j \in R \setminus S_{r'}$. Observe that by the fact that $\hat{sc}(j) < \frac{2n}{3(k+2)} < \frac{n}{(k+2)}$, it holds that at least $(|Z(S_{r'}, \texttt{dRep0})| - \frac{n}{(k+2)})$ voters from $Z(S_{r'}, \texttt{dRep0})$ are voting against j.

$$|Z(S_{r'+1}, \texttt{dRep0})| \geq \frac{(k+2-r')n}{(k+2)} - \frac{n}{(k+2)} = \frac{(k+2-(r'+1))n}{(k+2)}$$

Hence, Claim 7.7 indeed holds. ☒

We now continue with proving Lemma 7.6. We fix $r = k$ and a set $S_k \subseteq R$ of k proposals, in Claim 7.7. Note that by the discussion in Section 7.1, we know that $k \leq m_i$ for every voter i, and therefore $|R| \geq k$, so that we can choose such a set S_k. We then apply Claim 7.5 for the (coherent) subinstance induced by the voters in $Z(S_k, \texttt{dRep0})$ and the proposals in $C \setminus S_k$. This implies that either at least $\frac{|Z(S_k,\texttt{dRep0})|}{3}$ voters agree with dRep0 in at least $\lceil\frac{|C \setminus S_k|}{2}\rceil$ proposals of $C \setminus S_k$, or there is a proposal $j \in C \setminus S_k$ with $\hat{sc}(j) \geq \frac{|Z(S_k,\texttt{dRep0})|}{3}$. Using now Claim 7.7, we have that $\frac{|Z(S_k,\texttt{dRep0})|}{3} \geq \frac{2n}{3(k+2)}$, and thus the second case is infeasible, since we have assumed that $\hat{sc}(j) < \frac{2n}{3(k+2)}$, for any $j \in C$. Coming to the first case, the voters that agree with dRep0 in at least $\lceil\frac{|C \setminus S_k|}{2}\rceil$ proposals of $C \setminus S_k$, also agree with dRep0 in all proposals of S_k. This leads to a total agreement with dRep0 of at least

$$k + \left\lceil \frac{|C \setminus S_k|}{2} \right\rceil = k + \left\lceil \frac{m-k}{2} \right\rceil \geq \left\lceil \frac{m+k}{2} \right\rceil.$$

Therefore, for every voter $i \in Z(S_k, \texttt{dRep0})$, it holds that

$$d(i, \texttt{dRep0}) \leq m - \lceil \frac{m+k}{2} \rceil = \lfloor \frac{m-k}{2} \rfloor.$$

This leads to $|A(\texttt{dRep0})| \geq |Z(S_k, \texttt{dRep0})| \geq \frac{2n}{3(k+2)}$ due to Claim 7.7, and completes the proof of the lemma. ⊠

Applying Lemma 7.6 to the largest coherent set of the given instance immediately proves the statement of the theorem. □

An immediate but noteworthy corollary of Theorem 7.4 concerns majority agreement and coherent instances.

Theorem 7.4: For a single dRep, PROXY SELECTION admits an approximation ratio of $\min\left\{n, \frac{3n(k+2)}{2|S|}\right\}$, where S is the largest coherent set of voters.

Corollary 7.8 *For a single dRep,* PROXY SELECTION *for coherent instances and majority agreement admits an approximation ratio of* 3.

The attentive reader might have observed that for majority agreement and coherent instances, the general impossibility result of Theorem 7.3 does not apply. In that case, one might wonder what the best achievable approximation ratio is. To partially answer this question we offer the following result.

Theorem 7.3: For a single dRep and any $k > 0$, the approximation ratio of PROXY SELECTION is $\Omega(n)$.

Theorem 7.9 *Let* $\varepsilon > 0$. *For a single dRep,* PROXY SELECTION *does not admit a* $(1.6 - \varepsilon)$ *approximation, even for coherent instances and majority agreement.*

Proof. Consider an instance in which $N = \{v_1, v_2, \ldots, v_8\}$ and $C = \{c_1, c_2, c_3, c_4\}$. Suppose that $v_i(c_1) = 1$ and that $\hat{v}_i(c_1) = \perp$, for every voter $i \in N$. Furthermore, there is exactly one voter whose preferences with respect to proposals c_2, c_3, c_4 belongs in $\{110, 101, 011, 100, 010, 001\}$ and two voters that are voting for $\{111\}$. Note that in the current proof, for simplicity, we abuse the notation and use strings instead of ordered tuples to indicate voters' preferences. In this instance, $\texttt{opt}(P) = sc(c_1) = 8$ and $\hat{sc}(j) = 5, \forall j \in C \setminus \{c_1\}$. Therefore, direct voting cannot result in an approximation factor that is better than $\frac{8}{5} = 1.6$. We will prove that for any possible choice of advertised ballot of a dRep, and if $D = \{t\}$, then $sc(\texttt{win}(\hat{P}_D)) \leq 5$, which again results to the claimed approximation factor. Figure 7.1 will be helpful as an illustration of the arguments that are going to be used.

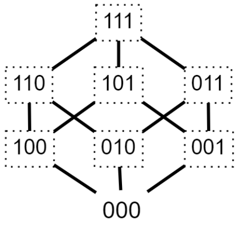


Figure 7.1: The graph illustrates the voters' preferences on the set of revealed proposals in the instance created for the proof of Theorem 7.9. Each boxed vertex represents the existence of a voter with the corresponding ballot, with respect to proposals c_2, c_3, c_4. If a dRep advertises the ballot of a vertex v in this graph, they will attract all voters whose preferences are within a distance of 1 from v.

- ➛ If $t = 111$, then $A(D)$ equals the set of voters whose preferences belong in $\{111, 110, 101, 011\}$. Therefore $|A(D)| = 5$ and hence $\hat{sc}_D(c_1) = 5$. However c_2 is both approved by the dRep and by a voter that doesn't belong to $A(D)$, consider, e.g., the voter who is voting for 100, which leads to $\hat{sc}_D(c_2) = 6$ and hence to a winning proposal $\texttt{win}(\hat{P}_D)$ such that $\hat{sc}_D(I') \geq 6$. Hence, $\texttt{win}(\hat{P}_D) \neq c_1$, and, $sc(\texttt{win}(\hat{P}_D)) = 5$.
- ➛ If $t = 110$, then $A(D)$ equals the set of voters whose preferences are $\{111, 110, 100, 010\}$. Therefore $|A(D)| = 5$ and hence $\hat{sc}_D(c_1) = 5$. Using the same rationale to before, one can observe that, again, $sc(\texttt{win}(\hat{P}_D)) = 5$. The proof is identical for $t = 101$ and $t = 011$.
- ➛ If $t = 100$, then $A(D)$ equals the set of voters whose preferences are $\{100, 110, 101\}$. Therefore $|A(D)| = 3$ and hence $\hat{sc}_D(c_1) = 3$. However c_2 is both approved by the dRep and by the voter whose preference vector is 111, who does not belong to $A(D)$, which leads to $sc(\texttt{win}(\hat{P}_D)) = 5$. The proof is identical for $t = 010$ and $t = 001$.
- ➛ If $t = 000$, then $|A(D)| = 3$, but $\hat{sc}_D(c_2) = 4$, which again leads to a winning proposal of $sc(\texttt{win}(\hat{P}_D)) = 5$. □

We now note that if we slightly relax our best-case scenario setting (see page 135) and consider arbitrary tie-breaking rules, then we can strengthen Theorem 7.9 to the following impossibility result.

Proposition 7.10 *Let $\varepsilon > 0$. For a single dRep,* PROXY SELECTION *does not admit a $(2 - \varepsilon)$-approximation if ties are broken in favor of the proposal with maximum revealed score, even for coherent instances and majority agreement.*

Proof. The proof is straightforward if we consider an instance of two candidate proposals, namely c_1 and c_2, and two voters whose preferences are depicted in the table below. Note that only voters' preferences with respect to c_2 are revealed to them.

	c_1	c_2
voter 1	1	1
voter 2	1	0

Obviously, $\mathtt{opt}(P) = 2$ and $\mathtt{opt}(\hat{P}) = 1$. Furthermore any dRep can attract at most one voter and hence, for any D such that $|D| = 1$, it holds that $\hat{sc}_D(c_1) \leq \hat{sc}_D(c_2)$. In case of ties, these are once again broken in favor of the proposal of maximum approval score in the revealed profile, i.e. in favor of c_2, since $\hat{sc}(c_2) > 0 = \hat{sc}(c_1)$. Hence, in any case, $\mathtt{win}(\hat{P}_D) = c_2$, and thus $sc(\mathtt{win}(\hat{P}_D)) = 1$, which equals $\frac{1}{2}\mathtt{opt}(P)$. □

Claim 7.11 *The instance used in Proposition 7.10 can be generalized towards proving that for any acceptance threshold $k > 0$, there exists an instance in which PROXY SELECTION does not admit a $(2-\epsilon)$-approximation, for any $\epsilon > 0$, if $\lambda \leq 2^{k-1}$, for arbitrary tie-breaking rule.*

Proof Sketch. Say that $\lambda = 2^{k-1}$ and consider a coherent instance, where $R = R_i$, for any voter i, with $n = 2\lambda$ voters and $m = R + 1$. Given the value of k, we define m such that $k = R - 1$. Voters' preferences follow:

- All voters approve proposal 1, which is the only proposal that does not belong to R.
- Only the first λ voters approve proposal 2.
- Only the 1st and 3rd group of $\frac{\lambda}{2}$ voters approve proposal 3.
- Only the 1st, 3rd, 5th, 7th group of $\frac{\lambda}{4}$ voters approve proposal 4.

  $\vdots$
- Only voters $\{1, 3, 5, \dots, n-1\}$ approve proposal $m - 1$.
- None of the voters approve proposal m.

Clearly, the optimal proposal has an approval score of 2λ. Furthermore, $\lceil \frac{R+k}{2} \rceil = \lceil \frac{2R-1}{2} \rceil = R$ and hence, any dRep, can attract at most one voter, and this can be done only by advertising that voter's ballot. As a consequence, introducing any set of dReps D, of size no more than λ, will result in $\hat{sc}_D(1) \leq \lambda$, but at the same time $\hat{sc}_D(2) = \hat{sc}(2) = \lambda$. Breaking-ties in an adversarial manner, a proposal of intrinsic score λ is being elected. □

Further generalizations of Claim 7.11 (such as handling different agreement threshold bounds or being independent of tie-breaking rules) are indeed possible, however we regard Claim 7.11 as an initial step towards understanding the limitations of the setting, and the scenario involving multiple dReps warrants a more comprehensive investigation.

To understand the power of coherence towards achieving good approximations, it is instructive to explore the limitations of the best possible dReps also on coherent instances. To this end, we provide a couple of results: the first generalizes Theorem 7.3 to be parameterized by the size of the largest coherent set, and the second shows robustness to coherent instances, as long as the agreement thresholds are sufficiently high. The take-away message of those results is that coherent sets are not a panacea, and can result in meaningful approximations only under further appropriate conditions.

Theorem 7.12 *Let $\varepsilon > 0$. For a single dRep and any $k > 0$,* PROXY SELECTION *does not admit an $(\frac{n}{|S|} - \epsilon)$-approximation, where S is the largest coherent set of voters in the instance.*

Proof. Recall that in the proof of Theorem 7.3, we designed an instance of PROXY SELECTION, in which every coherent set is formed by two voters, which implied inapproximability of $\frac{n}{2}$. At what follows, we will generalize that construction to instances with coherent sets of size up to any $r > 2$. We create $r/2$ copies of each voter (more precisely, $\lfloor r/2 \rfloor$ copies of each voter with an odd index and $\lceil r/2 \rceil$ copies of each voter with an even index). Clearly, the new number of voters is now $n' = n \cdot \frac{r}{2}$. The analysis from Theorem 7.3 directly carries over but now a proposal of score at most r gets elected instead of proposal m which has score n'. The maximum coherent set is of size r, and hence, the $\frac{n'}{r}$ inapproximability. □

Theorem 7.13 *Let $\varepsilon > 0$. For a single dRep and $k = m - 2c(m)$, with $c(m) \in o(m)$, the approximation ratio of* PROXY SELECTION *is $\Omega(n)$, even for coherent instances.*

Proof. We are going to construct an instance where all the voters form a single coherent set of size n. Moreover, $m_i = m - 1$ for all $i \in [n]$. Hence, we just write c rather than $c(m)$, i.e., $k = m - 1 - 2c$. We assume that $m = (n - 1)(c + 1) + 1$ and $n \geq 4$. In the created instance we have that

- for every voter $i \in [n]$, her preferences with respect to proposal m are as follows: $v_i(m) = 1$ and $\hat{v}_i(m) = \perp$,
- voter $i \in [n - 1]$ approves proposals $(i - 1)(c + 1) + 1, (i - 1)(c + 1) + 2, \dots, i(c + 1)$ and disapproves everything else (except for m, of course, but this does not belong to R_i, and the preference of voter n with respect to any proposal j satisfies $\hat{v}_i(j) = 1$.

Clearly, the optimal solution satisfies n voters. Similarly to the proof of Theorem 7.12, we claim that a single dRep, t, regardless of their advertised type, will contribute to electing a proposal approved by at most two voters. Towards this, we begin by showing that t cannot attract more than one voter. First note that, for $i, j \in [n - 1]$, we have $H(\hat{v}_i, \hat{v}_j) = 2c + 2$. Also, for $i \in [n-1]$, we have $H(\hat{v}_i, \hat{v}_n) = m-1-(c+1) \geq 3(c+1)+1-1-c-1 = 2c+2$, where we used both $m = (n-1)(c+1)+1$ and $n \geq 4$ for the inequality. Now, suppose that t attracts at least two distinct voters, say $i, j \in [n]$. Then,

$$2c + 2 \leq H(\hat{v}_i, \hat{v}_j) \leq d(i,t) + d(j,t) \leq \\ 2 \cdot \left\lfloor \frac{(m-1)-(m-1-2c)}{2} \right\rfloor = 2c,$$

which is a contradiction. We conclude that t may attract at most one voter.

If t does not attract any voters, then all voters vote directly and this leads to the election of a proposal approved by exactly two voters. On the other hand, if t attracts one voter, say i, we claim that t must have voted in favor of at least one proposal other than m. Indeed, if $t(j) = 0$ for every proposal j (except maybe for proposal m), then $d(i,t)$ would be at least $c + 1 > \left\lfloor \frac{(m-1)-(m-1-2c)}{2} \right\rfloor$ and they would not have attracted voter i.

As a result, a proposal of total approval 2 in the intrinsic profile is elected, instead of proposal m, and the $n/2$ inapproximability follows. □

We conclude our discussion for the case of one dRep with a complementary result of a computational nature: a theorem that establishes that finding a dRep to attract voters in a way that ultimately elects the optimal proposal is computationally hard. Consequently, PROXY SELECTION turns out to be challenging both from the standpoint of information theory and computational complexity.

Theorem 7.14 *The decision variant of* PROXY SELECTION *is NP-hard, even for majority agreement and a single dRep.*

Proof. We will establish NP-hardness for the decision version of PROXY SELECTION, where for some parameter r, we want to answer if there exists an advertised type for a dRep, so that the intrinsic score of the elected outcome is at least r. We will reduce from the problem MINIMAX APPROVAL VOTING (MAV) problem, which is a known NP-hard problem in voting theory [FL97; LMM07]. We note that the NP-hardness has been established for instances with m being even, and $\theta = m/2$. Given such an instance I, we create an instance I' of PROXY SELECTION as follows:

In MAV, we are given an instance I of m binary proposals and n ballots where $v_i \in \{0,1\}^m$, $i \in [n]$ and we are asked for a vector v for which it holds that $\max_{i\in[n]} H(v_i, v) \leq \theta$, where H is the Hamming distance between two vectors of the same size.

[FL97] Frances and Litman (1997): On Covering Problems of Codes.
[LMM07] LeGrand et al. (2007): Some Results on Approximating the Minimax Solution in Approval Voting.

- We have $m' = m + 3$ binary candidate proposals, i.e., three additional proposals from I: $\{c_1, \ldots, c_m, c_{m+1}, c_{m+2}, c_{m+3}\}$.
- We have n voters corresponding to the voters of I, and an additional number of $n + 1$ dummy voters, for a total of $2n + 1$ voters.
- For every voter $i \in [n]$, belonging to the group of the first n voters, their preferences for the first m proposals in I' are just as they are in I, and they are all revealed, so that $m_i = m$. The remaining three proposals are not

visible for these voters and their intrinsic preferences are that $v_i(c_{m+1}) = 1$, $v_i(c_{m+2}) = v_i(c_{m+3}) = 0$.

- For the dummy voters, none of them approve the first m proposals, which are also not revealed to them. As for the last three proposals, there are exactly two dummy voters, who will be referred to as the special dummy voters, who approve all three proposals, and all three are revealed to them. All the remaining $n-1$ dummy voters approve only the proposals c_{m+2} and c_{m+3}, which are revealed to them, whereas c_{m+1} is disapproved, and also not revealed to them.
- We set $r = n+2$ and $\lambda = 1$, i.e. we have only one dRep available. Hence we are looking for an advertised type of the dRep, so that the instrinsic score of the elected outcome is at least $n+2$.

An illustrative exposition of voters' ballots follows. In the table below, light gray cells correspond to preferences that are not revealed to the voters and dark gray cells correspond to preferences that are derived from I.

	m proposals	c_{m+1}	c_{m+2}	c_{m+3}
n **voters**	[···]	1	0	0
2 **special dummy voters**	0	1	1	1
$n-1$ **dummy voters**	0	0	1	1

Before we proceed, note that the only proposal that has an intrinsic score of $n+2$ is the proposal c_{m+1}, while all the others have lower scores. But c_{m+1} cannot be elected via only direct voting, since it is not revealed to the first n voters. Hence, the question is whether there exists an advertised type for the dRep that can make c_{m+1} elected.

For the forward direction, suppose that I is a YES-instance of MAV. Then there exists a vector $v \in \{0,1\}^m$ for which it holds that

$$\max_{i \in [n]} H(v_i, v) \leq m/2.$$

Consider now that the dRep advertises the type $t = (v, 1, 0, 0)$. The dRep will attract the first n voters, since they agree with them in at least $m/2$ of their revealed proposals. She will not attract any of the dummy voters, all of which disagree with them on two proposals. Therefore, the dRep will have a weight of n, and since the dummy voters vote directly, the proposal c_{m+1} will collect a score of n+2 and will be the winner of the election.

For the reverse direction, suppose that I is a NO-instance of MAV. Then for any possible vector $v \in \{0,1\}^m$, it holds that there exists at least one voter $i^* \in [n]$, such that $H(v_{i^*}, v) > m/2$. Fix now such an arbitrary vector

$v \in \{0,1\}^m$. We will consider all possible cases for the advertised type of the dRep, t, and show that c_{m+1} cannot get elected, i.e, I' is a NO-instance. We use a natural tie-breaking rule that resolves any tie in favor of the proposal of maximum approval score in the revealed profile. Therefore, in all the cases below, any tie that involves proposals $c_{m+1}, c_{m+2}, c_{m+3}$ is broken against c_{m+1} due to the fact that $\hat{sc}(c_{m+1}) < \hat{sc}(c_{m+2}) = \hat{sc}(c_{m+3})$.

- Suppose that $t = (v, 0, x, y)$, for any $x, y \in \{0,1\}$. It is easy to see that since the dRep does not approve c_{m+1}, it is not possible that this proposal wins the election.
- Suppose that $t = (v, 1, 0, 0)$. In this case, the dRep does not attract any of the dummy voters. Hence, the proposals c_{m+2} and c_{m+3} receive a score of $n+1$. As for proposal c_{m+1}, it is crucial to note that since I is a NO-instance of MAV, t can attract at most $n-1$ of the first n voters. Therefore together with the two special dummy voters, the proposal c_{m+1} will have a score of at most $n+1$. By the tie-breaking rule that we have assumed, this means that the winner of the election will be either c_{m+2} or c_{m+3}.
- Suppose that $t = (v, 1, 1, 0)$, or $t = (v, 1, 1, 1)$, or $t = (v, 1, 0, 1)$. Without loss of generality, we analyze the former. In this case, the dRep attracts all dummy voters. Hence, the dRep has a weight of at least $n+1$ and possibly more by some of the first n voters who delegate to her. Hence, the winner will either be some proposal that is approved by v and also has the highest approval rate among the voters who did not delegate to dRep, or there is a tie among all the proposals approved by t. By the tie-breaking rule, it is not possible that c_{m+1} is elected.

We have established that no matter what the dRep advertises, it is impossible that c_{m+1} is elected, and hence I' is a NO-instance, which concludes the proof. □

We conclude by noting that, importantly, all of the approximation guarantees presented in the current chapter can be obtained in polynomial time.

7.2.1 Multiple dReps

We turn our attention to the case of multiple dReps ($\lambda \geq 2$), as this is not captured by the impossibility of Theorem 7.3. Is it perhaps possible to achieve much better approximations by using sufficiently many dReps? A reinforcing observation is that for majority agreement, 2 dReps suffice to elect the optimal proposal.

Theorem 7.3: For a single dRep and any $k > 0$, the approximation ratio of PROXY SELECTION is $\Omega(n)$.

Theorem 7.15 *When $\lambda = 2$, PROXY SELECTION for majority agreement can be optimally solved.*

Proof. We begin by showing that there exists a ballot type, the advertisement of which, results to an attraction of at least half of the voters that vote in favor of the optimal proposal. Recall that

$$\texttt{dRep0}(j) = \begin{cases} 1, \text{ if } j = \texttt{win}(P), \\ 0, \text{ otherwise.} \end{cases} \quad \texttt{dRep1}(j) = 1, \forall j \in C.$$

We will make use of the following proposition. Note that dRep0 and dRep1 are not the "all-1s" and the "all-0s" dReps, but rather they always vote 1 for the optimal proposal. Therefore, the proof of the proposition, although not involved, is not immediate.

Lemma 7.16 *In an instance of* PROXY SELECTION, *with $k = 0$, it holds that* $\max\{|A(\texttt{dRep0})|, |A(\texttt{dRep1})|\} \geq \frac{|N'|}{2}$, *where* $N' = \{i \in N : {}_{v_i(\texttt{win}(P))=1}\}$.

Proof of Lemma 7.16. We will prove that any arbitrary voter $i \in N'$ belongs to at least one of $A(\texttt{dRep0})$ and $A(\texttt{dRep1})$. We focus on the proposals $R'_i = R_i \setminus \{\texttt{win}(P)\}$. Let $R'_i(0) = |\{j \in R'_i : \hat{v}_i(j) = 0\}|$, and similarly, $R'_i(1) = |\{j \in R'_i : \hat{v}_i(j) = 1\}|$. Then, trivially, it holds that either $R'_i(0) \geq R'_i(1)$ or $R'_i(1) > R'_i(0)$. If $\texttt{win}(P) \in R_i$, voter i agrees with both dRep0 and dRep1 on $\texttt{win}(P)$, whereas if $\texttt{win}(P) \notin R_i$, by the definition of the distance function, $\texttt{win}(P)$ does not affect the distance of voter i from any dRep. Hence in both cases, proposal $\texttt{win}(P)$ does not contribute to the distance of i from the two dReps, and therefore,

$$\min\{d(i, \texttt{dRep0}), d(i, \texttt{dRep1})\} \leq 0 + \left\lfloor \frac{|R'_i|}{2} \right\rfloor \leq \left\lfloor \frac{|R_i|}{2} \right\rfloor,$$

where the minimum is achieved by dRep0 if $R'_i(0) \geq R'_i(1)$, and by dRep1 otherwise. ☒

In fact, the proof of Lemma 7.16 implies something stronger, namely that in any instance of PROXY SELECTION for majority agreement, each voter that, secretly or not, approves the optimal proposal, belongs either to $A(\texttt{dRep0})$ or to $A(\texttt{dRep1})$ (or both). Therefore, a direct consequence is that if $D = \{\texttt{dRep0}, \texttt{dRep1}\}$ then $A(D) = N'$. Hence, if $\lambda = 2$, we can retrieve the optimal solution, since it would hold that $\hat{sc}_D(\texttt{win}(P)) = |N'|$, while $\hat{sc}_D(j) \leq |N'|, \forall j \in C$. □

But what about instances in which voters are more discerning, indicated by larger values of k? Whether good approximations with multiple delegation representatives are achievable *in general* is still to be determined. We begin with the case of coherent instances and we provide the following theorem, which suitably generalizes Theorem 7.15.

Theorem 7.15: When $\lambda = 2$, PROXY SELECTION for majority agreement can be optimally solved.

Theorem 7.17 *When* $\lambda = 2^{k+1}$, *PROXY SELECTION for coherent instances and any* $k \geq 0$ *can be optimally solved.*

Proof. Let R be the set of commonly revealed proposals to the voters of the given instance. It is without loss of generality here to assume that $\mathtt{win}(P) \notin R$, or otherwise, the direct voting would result in the election of $\mathtt{win}(P)$. To create a set of dReps D, we fix an arbitrary set $S_k \subseteq R$, of k proposals, and for every possible binary vector on S_k, i.e., for every $\sigma \in 2^{S_k}$, we add to D exactly two dReps, namely $t_{\sigma,0}$ and $t_{\sigma,1}$, advertising the following, with respect to a proposal j:

$$t_{\sigma,0}(j) = \begin{cases} \sigma(j), & \text{if } j \in S_k, \\ 1, & \text{if } j = \mathtt{opt}(P), \\ 0, & \text{otherwise.} \end{cases} \quad t_{\sigma,1}(j) = \begin{cases} \sigma(j), & \text{if } j \in S_k, \\ 1, & \text{otherwise.} \end{cases}$$

To prove the statement, it is sufficient, to show that $A(D) = N$, i.e., that D can attract all voters from N. We fix an arbitrary voter $i \in N$. Definitely, there is a vector, say σ', that defines a pair of dReps in D, say $t_{\sigma',0}$ and $t_{\sigma',1}$ (henceforth denoted by t_1 and t_2), such that i totally agrees in all proposals of S_k both with t_1 and t_2. Formally, if for a given vector x and a set of proposals Y, we denote by $x_{|Y}$ the projection of x to the proposals in Y, the following holds:

$$\max\{H(\hat{v}_{i|S_k}, t_{1_{|S_k}}), H(\hat{v}_{i|S_k}, t_{2_{|S_k}}\} = 0 \tag{7.1}$$

Let $R'_i := R_i \setminus S_k = R \setminus S_k$ and for $z \in \{0,1\}$ we define

$$R'_i(z) := |\{j \in R'_i : \hat{v}_i = z\}|.$$

Then, either $R'_i(0) \geq R'_i(1)$, or $R'_i(1) > R'_i(0)$. Therefore,

$$\min\{H(\hat{v}_{i|R'_i}, t_{1|R'_i}), H(\hat{v}_{i|R'_i}, t_{2|R'_i}\} \leq \left\lfloor \frac{|R'_i|}{2} \right\rfloor. \tag{7.2}$$

Combining Equations (7.1) and (7.2), we have that voter i agrees either with t_1 or with t_2, in a number of proposals that is at least

$$k + \left\lceil \frac{|R'_i|}{2} \right\rceil = k + \left\lceil \frac{m_i - k}{2} \right\rceil \geq \left\lceil \frac{m_i + k}{2} \right\rceil.$$

Consequently, every voter will delegate to a dRep from D and the optimal proposal will be elected. □

Theorem 7.17 provides a bound on the sufficient number of dReps required to make sure that the optimal proposal is elected and raises a question regarding positive results (both optimal and approximate) for not-necessarily-coherent instances. In Theorem 7.21 below we provide a generalized and more refined version of this result, that relates the achievable approximation with the required number of dReps and the parameters of the instance, but does not need to assume that the instances are coherent.

Theorem 7.21: When $\lambda = \min\{n, \zeta 2^{k+1}\}$, PROXY SELECTION admits an approximation ratio of $\frac{\gamma}{3\zeta}$, where γ is the minimum number of $(k, m-k)$-coherent sets that can form a partition of N, and $\zeta \leq \gamma$ with $\zeta \in \mathbb{N}$.

7.2.2 Beyond Coherent Instances

Coherence has been proven to be very useful towards achieving meaningful approximation guarantees. At what follows, we define a more refined notion, namely a quantified version of it, which provides further insights into the structure of instances and how these affect the achievable approximations. In particular, we use the notion of (x, δ)-coherent sets for sets of voters that have a common set of proposals of size x in their revealed sets, as well as at most δ additional proposals.

Definition 7.3 *A set of voters $N' \subseteq N$ is called (x, δ)-coherent if there exists a set $X \subseteq C$ such that for every $i \in N$ the following hold: $X \subseteq R_i$, $|X| \geq x$, and $|R_i \setminus X| \leq \delta$.*

Using Definition 7.3, we generalize the result of Theorem 7.4, with an additional loss in the factor that is dependent on the type of (x, δ)-coherent sets that an instance admits.

Theorem 7.4: For a single dRep, PROXY SELECTION admits an approximation ratio of $\min\left\{n, \frac{3n(k+2)}{2|S|}\right\}$, where S is the largest coherent set of voters.

Theorem 7.18 *For a single dRep and for any $\delta > 0$, PROXY SELECTION admits an approximation ratio of* $\min\left\{n, \frac{3n(k+\delta+2)}{2|S|}\right\}$*, where S is the largest $(k+\delta, \delta)$-coherent set in the instance.*

Proof. By electing any proposal, one can straightforwardly obtain an approximation factor of n. For the remaining of the proof, say that $\frac{2n(k+\delta+2)}{2|S|} < n$. Additionally, for now and for ease of exposition, suppose that for every voter $i \in N$, it holds that $v_i(\mathtt{win}(P)) = 1$, $\hat{v}_i(\mathtt{win}(P)) = \perp$. At the end of the proof we will show that this assumption can be dropped. For the studied case, we will prove an approximation ratio of $\frac{3n(k+\delta+2)}{2|S|}$.

The proof follows the same rationale as the proof of Theorem 7.4, which was based on applying Lemma 7.6 to the largest coherent set of the instance. Similarly, to prove Theorem 7.18, it suffices to prove an analogous lemma, that we state below. Recall that

$$\mathtt{dRep0}(j) = \begin{cases} 1, \text{ if } j = \mathtt{win}(P), \\ 0, \text{ otherwise.} \end{cases} \qquad \mathtt{dRep1}(j) = 1, \forall j \in C.$$

Lemma 7.6: In a coherent instance of PROXY SELECTION, either $sc(\mathtt{win}(\hat{P})) \geq \frac{2n}{3(k+2)}$, or $|A(\mathtt{dRep0})| \geq \frac{2n}{3(k+2)}$. Hence, an approximation factor of $\frac{3(k+2)}{2}$ can be achieved.

Lemma 7.19 *In an instance of* PROXY SELECTION, *with N being $(k+\delta,\delta)$-coherent, it either holds that $sc(\texttt{win}(\hat{P})) \geq \frac{2n}{3(k+\delta+2)}$ or $|A(\texttt{dRep0})| \geq \frac{2n}{3(k+\delta+2)}$.*

Proof of Lemma 7.19. Suppose that there exists a proposal $j' \in C$ such that $\hat{sc}(j') \geq \frac{2n}{3(k+\delta+2)}$. Then the proof follows by the fact that

$$sc(win(\hat{P})) \geq \hat{sc}(win(\hat{P})) \geq \hat{sc}(j').$$

So we can assume that for every proposal $j \in C$, it holds that $\hat{sc}(j) < \frac{2n}{3(k+\delta+2)}$. Next, we state a claim which is a generalization of Claim 7.7. Its proof is almost identical to the proof of Claim 7.7.

Claim 7.7: Consider a coherent instance, with R being the set of proposals commonly revealed to all voters. Suppose also that $\hat{sc}(j) < \frac{2n}{3(k+2)}$ for any $j \in R$. For any $r \in [k]$ and for any set $S_r \subseteq R$ of r proposals, let $Z(S_r, \texttt{dRep0})$ be the set of voters that totally agree with $\texttt{dRep0}$ in all proposals of S_r. Then $|Z(S_r, \texttt{dRep0})| \geq \frac{(k+2-r)n}{(k+2)}$.

Claim 7.20 *Consider an instance where the set of voters N forms a $(k+\delta,\delta)$-coherent set, for some $\delta \geq 0$. Suppose also that $\hat{sc}(j) < \frac{2n}{3(k+\delta+2)}$, for any $j \in C$. Let X be the set of commonly revealed proposals to all voters, which by definition satisfies $|X| \geq k+\delta$. For any set $S_r \subseteq X$ of r proposals, with $r \in [k+\delta]$, let $Z(S_r, \texttt{dRep0})$ be as defined in Claim 7.7. Then $|Z(S_r, \texttt{dRep0})| \geq \frac{(k+2+\delta-r)n}{(k+\delta+2)}$.*

We proceed with proving Lemma 7.19, using Claim 7.20. Fix $r = k+\delta$ in Claim 7.20, and let $S_r \subseteq X$ be a set of $k+\delta$ proposals. Let also $C' = X \setminus S_r$. We examine the number of voters who will eventually delegate to $\texttt{dRep0}$.

Theorem 7.9: Let $\varepsilon > 0$. For a single dRep, PROXY SELECTION does not admit a $(1.6{-}\varepsilon)$ approximation, even for coherent instances and majority agreement.

We observe that every proposal of C' belongs to the revealed set of any voter of $Z(S_r, \texttt{dRep0})$. Therefore, we can apply Corollary 7.8 for $Z(S_r, \texttt{dRep0})$ and C'. This implies that either there exists a proposal in C' approved by $\frac{Z(S_r,\texttt{dRep0})}{3}$ voters or at least $\frac{Z(S_r,\texttt{dRep0})}{3}$ voters agree with $\texttt{dRep0}$ on at least $\lceil\frac{|C'|}{2}\rceil$ proposals. By Claim 7.20, we know that $|Z(S_r, \texttt{dRep0})| \geq \frac{2n}{k+2+\delta}$. Therefore, by the assumption we have made, it is not possible to have a proposal with a score of $\frac{|Z(S_r,\texttt{dRep0})|}{3}$, hence, at least $\frac{2\cdot|Z(S_r,\texttt{dRep0})|}{3(k+\delta+2)}$ voters of $Z(S_r, \texttt{dRep0})$ agree with $\texttt{dRep0}$ in at least $\lceil\frac{|C'|}{2}\rceil$ proposals from C'. But, by the definition of $Z(S_r, \texttt{dRep0})$, they also agree on the $k+\delta$ proposals of S_r. In total, their agreement with $\texttt{dRep0}$ is at least:

$$k+\delta+\left\lceil\frac{|C'|}{2}\right\rceil \geq k+\delta+\left\lceil\frac{m_i-k-\delta-\delta}{2}\right\rceil \geq \left\lceil\frac{m_i+k}{2}\right\rceil,$$

where the first inequality holds due to the fact that $|C'| = |X| - (k+\delta)$, and $m_i \leq |X| + \delta$. As a consequence all these voters, i.e., at least $\frac{2n}{3(k+\delta+2)}$ in number, will delegate to $\texttt{dRep0}$. ☒

Hence, Lemma 7.19 holds and Theorem 7.18 follows, under the assumption that all voters secretly approve the optimal proposal. Suppose that in a given instance the assumption doesn't hold. Focusing on the set of voters, called N', that satisfy the assumption and applying the described procedure, one can prove that either there is a proposal of score at least $\frac{2|N'|}{3(k+\delta+2)}$ or dRep0 attracts at least $\frac{2|N'|}{3(k+\delta+2)}$ voters. Given that $sc(\texttt{win}(P)) = |N'|$, an approximation ratio of $\frac{3(k+\delta+2)}{2}$ is implied.

To conclude the proof, we note that the desired approximation ratio holds true by focusing on the set S, just like in Theorem 7.4. □

The main result of the subsection is a relaxation of Theorem 7.17. Unlike Theorem 7.17, the result that follows does not require any structural assumptions, and relates the approximation with the number of dReps, the threshold bound and the structure of the instance in terms of approximate coherence.

Theorem 7.17: When $\lambda = 2^{k+1}$, PROXY SELECTION for coherent instances and any $k \geq 0$ can be optimally solved.

Theorem 7.21 *When $\lambda = \min\{n, \zeta 2^{k+1}\}$, PROXY SELECTION admits an approximation ratio of $\frac{\gamma}{3\zeta}$, where γ is the minimum number of $(k, m-k)$-coherent sets that can form a partition of N, and $\zeta \leq \gamma$ with $\zeta \in \mathbb{N}$.*

Proof. For ease of exposition we assume that for every voter $i \in N$, it holds that $v_i(\texttt{win}(P)) = 1$, $\hat{v}_i(\texttt{win}(P)) = \perp$. We note that this assumption is without any loss in the approximation factor, likewise in the proof of Theorem 7.18 in which we proved that focusing only on the set of voters that are secretly in favor of the optimal proposal does not affect the ratio.

Trivially, if we set D to consist of one dRep t_i for every voter v_i such that

$$t_i(j) = \begin{cases} 1\,, \text{if } j = \texttt{win}(P) \\ v_i(j)\,, \text{otherwise} \end{cases}$$

then $A(D) = N$ and hence $\texttt{win}(\hat{P}_D) = \texttt{win}(P)$. Therefore, with n dReps we can retrieve the optimal solution and, as a consequence, the claimed approximation factor holds. We will now proceed with proving that whenever $\zeta 2^{k+1} < n$, we can use a set of dReps D, where $|D| \leq \zeta 2^{k+1}$, to elect a proposal j such that $sc(j) \geq \frac{3\zeta}{\gamma}\texttt{opt}(P)$.

We proceed with the following lemma, the proof of which is analogous to the proof of Theorem 7.17.

Lemma 7.22 *In an instance of PROXY SELECTION, in which N can be partitioned in at most γ $(k, m-k)$-coherent sets, PROXY SELECTION can be optimally solved if $\lambda = \gamma 2^{k+1}$.*

Using 2^{k+1} dReps for any of the sets described in the statement of the theorem, as indicated by Lemma 7.22, we can create a set of dReps D such that $|D| \leq \gamma 2^{k+1}$ and $A(D) = N$. This proves the statement of the theorem for $\zeta = \gamma$. Fix now a number $\zeta < \gamma$. Among the γ sets produced by Lemma 7.22, let us focus on the ζ sets of largest size. Let also $N' \subseteq N$ be the voters in these sets. Then we can create a set of dReps $D' \subseteq D$, such that $|D'| \leq \zeta 2^{k+1}$ and $A(D') \supseteq N'$. But then, $|A(D')| \geq |N'|$.

Although all dReps of D' vote in favor of $\texttt{win}(P)$ and this can indeed be the winning proposal of profile $P_{D'}$, it may also be the case that $\texttt{win}(P_{D'}) \neq \texttt{win}(P)$. In fact, if some dReps from D' that are voting in favor of $\texttt{win}(P_{D'})$, attract voters that vote against $\texttt{win}(P_{D'})$, the intrinsic score of $\texttt{win}(P_{D'})$ may significantly differ from $sc(\texttt{win}(P_{D'}))$. To avoid this behaviour we suggest to create a new set of dReps, say D'', by deleting from D' all dReps of the form $t_{\sigma,1}$, i.e., all dReps that are voting in favor of proposals that do not belong in S_k, as defined in the proof of Theorem 7.17. Interestingly $|D''| = \frac{|D'|}{2}$. We will now compute $|A(D'')|$.

Suppose first that the number of voters attracted by dReps in $D' \setminus D''$ is at most $\frac{2|N'|}{3}$. Then, $|A(D'')| \geq |N'| - \frac{2|N'|}{3} = \frac{|N'|}{3}$. We define $\alpha := |\cup_{i\in\tilde{N}} R_i|$ and $\beta := \min\{|R_i|, i \in \tilde{N}\}$, where $\tilde{N}$ is any of the γ (k,m-k)-coherent sets of the voters' partition, as described in the statement of the theorem. We call N_1 the number of voters, attracted by dReps in $D' \setminus D''$ and say that $N_1 \geq \frac{2|N'|}{3}$. But then, for every voter $i \in N_1$ it holds that $R_i(1) \geq \frac{|R_i|}{2} \geq \frac{\beta}{2}$. But then, the total number of approvals in $\hat{P}$ for proposals in $C \setminus S_k$, equals $\sum_{i\in N_1} R_i(1) \geq |N_1|\frac{\beta}{2} \geq \frac{2|N'|}{3}\frac{\beta}{2} = \frac{|N'|\beta}{3}$. All these approvals are spread between α proposals. Therefore, there exists a proposal $j \in R$ for which $\widehat{sc}(j) \geq \frac{|N'|\beta}{3\alpha}$.

Consequently, either there is a proposal approved by $\frac{|N'|\beta}{3\alpha}$ voters or there exists a set of dReps D'' that can attract $\frac{|N'|\beta}{3\alpha}$ voters without voting in favor of a proposal disapproved by a voter in $A(D'')$. Using the fact that $|N'| \geq \frac{\zeta}{\gamma}n \geq \frac{\zeta}{\gamma}\texttt{opt}(P)$, we have an approximation ratio of $\frac{\gamma}{\zeta} \cdot \frac{\beta}{3\alpha}$. Obviously, if every of the considered ζ, out of the γ (k,m-k)-coherent sets, is coherent, then an approximation ratio of $\frac{\gamma}{3\zeta}$ is implied. □

An interesting corollary of Theorem 7.21 (which could also be deduced from the proof of Theorem 7.17) is the following: When aiming for an optimal solution with 2^{k+1} dReps, it's not a necessity for instances to be coherent; rather, the key factor is the existence of a set of k proposals commonly revealed to all voters.

We conclude the discussion on generalizations with the following theorem, for the case of majority agreement ($k = 0$), which generalizes Corollary 7.8 by relating the achievable approximation to the structure of the revealed sets, once again without the requirement of coherence.

Corollary 7.8: For a single dRep, PROXY SELECTION for coherent instances and majority agreement admits an approximation ratio of 3.

Theorem 7.23 *For a single dRep,* PROXY SELECTION *for majority agreement admits an approximation ratio of* $\min\{n, \frac{3\alpha}{\beta}\}$*, where* $\alpha := |\cup_{i\in N} R_i|$ *and* $\beta := \min\{|R_i|, i \in N\}$.

Proof. It is apparent that any instance of PROXY SELECTION, is approximable within a factor of n, since it holds that electing arbitrarily a proposal $j \in C$ results in a winning proposal that satisfies $sc(j) \geq 1 \geq \frac{1}{n}\texttt{opt}(P)$. Therefore suppose that $\min\{n, \frac{3\alpha}{\beta}\} = \frac{3\alpha}{\beta}$. From Lemma 7.16 it holds that either `dRep0` or `dRep1` can attract at least half the electorate, where

$$\texttt{dRep0}(j) = \begin{cases} 1, \text{ if } j = \texttt{win}(P), \\ 0, \text{ otherwise.} \end{cases} \qquad \texttt{dRep1}(j) = 1, \; \forall j \in C.$$

If this holds for $D = \{\texttt{dRep0}\}$, we are done, because either the optimal proposal wins the election, via `dRep0`, or another proposal wins through the voters who vote directly, in which case the intrinsic score of the winning proposal j is at least $\widehat{sc}_D(j) \geq A(D) \geq \frac{n}{2} \geq \frac{n}{3} \geq \frac{n\beta}{3\alpha} = \frac{\beta}{3\alpha}\texttt{opt}(P)$. Otherwise, if `dRep1` attracts at least half the electorate, say that all voters from a set $N_1 \subseteq N$ are being attracted by `dRep1`. Then, for every voter $i \in N_1$, let $R_i(0) = |\{j \in R_i : \hat{v}_i = 0\}|$ and $R_i(1) = |\{j \in R_i : \hat{v}_i = 1\}|$. It should hold that $R_i(1) > R_i(0)$, due to the fact that $i \in A(\texttt{dRep0})$.

- Suppose that $|N_1| \geq \frac{2n}{3}$, For every voter $i \in N_1$ it holds that $R_i(1) \geq \frac{|R_i|}{2} \geq \frac{\beta}{2}$. But then, the total number of approvals in $\hat{P}$ equals

  $$\sum_{i\in N_1} R_i(1) \geq |N_1|\frac{\beta}{2} \geq \frac{2n}{3}\frac{\beta}{2} = \frac{n\beta}{3}.$$

  All these approvals are spread between α proposals. Therefore, there exists a proposal $j \in R$ for which $\widehat{sc}(j) \geq \frac{n\beta}{3\alpha}$.

- Suppose that $|N_1| < \frac{2n}{3}$. Then, the set of voters in $N \setminus N_1$ should be attracted by `dRep0`, again by Lemma 7.16. In total, these are at least $n - \frac{2n}{3} = \frac{n}{3}$. Therefore, if $D = \{\texttt{dRep0}\}$, the delegation representative will vote with a weight of at least $\frac{n}{3}$, only in favor of $\texttt{win}(P)$, which would result in $\widehat{sc}_D(\texttt{win}(P)) \geq \frac{n}{3}$. Then, either $\texttt{win}(P)$ will win the election achieving an optimal solution, or a proposal $j \in C \setminus \{\texttt{win}(P)\}$ will win. In the later case, it should hold that $\widehat{sc}_D(j) \geq \widehat{sc}_D(\texttt{win}(P)) \geq \frac{n}{3} \geq \frac{n\beta}{3\alpha}$. However, $\texttt{dRep0}(j) = 0$, and, $\widehat{sc}_D(j) = \widehat{sc}(j)$.

Whenever $\texttt{win}(\hat{P}_D) \neq \texttt{win}(P)$, there is a proposal $j \in C$ such that $\widehat{sc}(j) \geq \frac{n\beta}{3\alpha} \geq \frac{\beta}{3\alpha} \cdot \texttt{opt}(P)$. The fact that $sc(j) \geq \widehat{sc}(j)$, concludes the proof. □

7.3 Concluding Discussion and Future Directions

In this chapter, we proposed and studied a model for proxy voting where the (less-informed) voters delegate their votes to the (fully-informed) proxies (dReps), once a certain agreement between their ballots is reached, or they vote directly otherwise. Our findings encompass essential insights into comprehending what is possible (potential) and what is not (limitations) in this setting. By identifying structural properties and other restrictions, we managed to escape the strong impossibilities that we established.

The upper and lower bounds presented are not always tight, and future work could focus on sharpening these bounds. Perhaps more interesting is the migration from the "best-case scenario" setting that we study. This would most probably entail the following two components:

- *An information model for the dReps*. It would be reasonable to assume that each delegate representative is correctly-informed about each voter i's approval preference of each proposal j with some probability p_{ij}, or that they are (perfectly or imperfectly) informed about a randomly-chosen set of proposals for each voter.
- *A rationality model for the dReps*. Delegate representatives might not have any incentives to coordinate towards the socially-desirable outcome, and they would need to be properly incentivized to do that, e.g., via the form of payments. Additionally, dReps could even have their own preferences regarding the proposals under consideration.

One could think of many other examples or refinements of the above, and the appropriate choice of information/rationality model for the dReps would depend on the application at hand. Tie-breaking rules that do not necessarily favor the optimal proposal also worth studying. Regardless of these choices however, the results of the "best-case scenario" should be the starting point of any investigation into those settings.

Besides those extensions, other directions that we see as promising routes for further research on the topic include different distance metrics, different voting rules, the multi-winner setting, elections on interdependent proposals, as well as a partial-delegation setting where voters can opt to delegate only on proposals for which they have no opinion.

Final Words

Bibliography

[AAL+19] Manel Ayadi, Nahla Ben Amor, Jérôme Lang, and Dominik Peters. *Single Transferable Vote: Incomplete Knowledge and Communication Issues*. In: *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 2019, pp. 1288–1296. [↷ p. 137.]

[ABC+17] Haris Aziz, Markus Brill, Vincent Conitzer, Edith Elkind, Rupert Freeman, and Toby Walsh. *Justified Representation in Approval-Based Committee Voting*. In: *Social Choice and Welfare* 2.48 (2017), pp. 461–485. [↷ p. 62.]

[ABE+19] Haris Aziz, Felix Brandt, Edith Elkind, and Piotr Skowron. *Computational Social Choice: The First Ten Years and Beyond*. In: *Computing and Software Science: State of the Art and Perspectives* (2019), pp. 48–65. [↷ p. 2.]

[ABE+22] Shiri Alouf-Heffetz, Laurent Bulteau, Edith Elkind, Nimrod Talmon, and Nicholas Teh. *Better Collective Decisions via Uncertainty Reduction*. In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. 2022, pp. 24–30. [↷ p. 137.]

[ABL+15] Georgios Amanatidis, Nathanaël Barrot, Jérôme Lang, Evangelos Markakis, and Bernard Ries. *Multiple Referenda and Multiwinner Elections Using Hamming Distances: Complexity and Manipulability*. In: *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 2015, pp. 715–723. [↷ p. 20.]

[ACM+05] Amit Agarwal, Moses Charikar, Konstantin Makarychev, and Yury Makarychev. *$O(\sqrt{logn})$ Approximation Algorithms for MIN UNCUT, MIN 2-CNF DELETION and DIRECTED CUT problems*. In: *Proceedings of the ACM Symposium on Theory of Computing (STOC)*. 2005, pp. 573–581. [↷ pp. 36, 37.]

[ADL22] Fatma Aslan, Hayrullah Dindar, and Jean Lainé. *When are Committees of Condorcet Winners Condorcet Winning Committees?* In: *Review of Economic Design* 26.3 (2022), pp. 417–446. [↷ p. 62.]

[AEG+11] Stéphane Airiau, Ulle Endriss, Umberto Grandi, Daniele Porello, and Joel Uckelman. *Aggregating Dependency Graphs into Voting Agendas in Multi-Issue Elections*. In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. 2011, pp. 18–23. [↷ p. 20.]

[AEH+18] Haris Aziz, Edith Elkind, Shenwei Huang, Martin Lackner, Luis Sánchez-Fernández, and Piotr Skowron. *On the Complexity of Extended and Proportional Justified Representation*. In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 2018, pp. 902–909. [↷ p. 62.]

[AFL+24] Georgios Amanatidis, Aris Filos-Ratsikas, Philip Lazos, Evangelos Markakis, and Georgios Papasotiropoulos. *On the Potential and Limitations of Proxy Voting: Delegation with Incomplete Votes*. In: *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 2024. [↷ p. 14.]

[AFS+21] Elliot Anshelevich, Aris Filos-Ratsikas, Nisarg Shah, and Alexandros Voudouris. *Distortion in Social Choice Problems: The First 15 Years and Beyond*. In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. 2021, pp. 4294–4301. [↷ p. 137.]

[AFV+21] Elliot Anshelevich, Zack Fitzsimmons, Rohit Vaish, and Lirong Xia. *Representative Proxy Voting*. In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 2021, pp. 5086–5093. [↷ p. 87.]

[ALS91] Stefan Arnborg, Jens Lagergren, and Detlef Seese. *Easy Problems for Tree-Decomposable Graphs*. In: *Journal of Algorithms* 12.2 (1991), pp. 308–340. [↝ pp. 95–99.]

[AM19] Ben Abramowitz and Nicholas Mattei. *Flexible Representative Democracy: An Introduction with Binary Issues*. In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. 2019, pp. 3–10. [↝ p. 85.]

[Ans76] Gertrude Elizabeth Margaret Anscombe. *On Frustration of the Majority by Fulfilment of the Majority's Will*. In: *Analysis* 36.4 (1976), pp. 161–168. [↝ p. 62.]

[AS21] Haris Aziz and Nisarg Shah. *Participatory Budgeting: Models and Approaches*. In: *Pathways Between Social Science and Computational Social Science: Theories, Methods, and Interpretations*. Springer, 2021, pp. 215–236. [↝ p. 134.]

[ASS10] Kenneth Arrow, Amartya Sen, and Kotaro Suzumura. *Handbook of Social Choice and Welfare*. Elsevier, 2010. [↝ p. 2.]

[AZ05] Adi Avidor and Uri Zwick. *Approximating MIN 2-SAT and MIN 3-SAT*. In: *Theory of Computing Systems* 38.3 (2005), pp. 329–345. [↝ p. 35.]

[BBC+11] Paolo Boldi, Francesco Bonchi, Carlos Castillo, and Sebastiano Vigna. *Viscous Democracy for Social Networks*. In: *Communications of the ACM* 54.6 (2011), pp. 129–137. [↝ pp. 85, 108.]

[BBD+04] Craig Boutilier, Ronen Brafman, Carmel Domshlak, Holger Hoos, and David Poole. *CP-Nets: A Tool for Representing and Reasoning with Conditional Ceteris Paribus Preference Statements*. In: *Journal of Artificial Intelligence Research* 21 (2004), pp. 135–191. [↝ p. 20.]

[BBR+22] Matthias Bentert, Niclas Boehmer, Maciej Rymar, and Henri Tannenberg. *Who Won? Winner Determination and Robustness in Liquid Democracy*. In: *CoRR (arXiv:2205.05482)* (2022). [↝ p. 109.]

[BCE+16] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel Procaccia. *Handbook of Computational Social Choice*. Cambridge University Press, 2016. [↝ pp. 2, 36.]

[BCE12] Felix Brandt, Vincent Conitzer, and Ulle Endriss. *Computational Social Choice*. In: *Multiagent Systems* 2 (2012), pp. 213–284. [↝ p. 2.]

[BD15] Dorothea Baumeister and Sophie Dennisen. *Voter Dissatisfaction in Committee Elections.* In: *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 2015, pp. 1707–1708. [↝ p. 137.]

[BDD+21] Ruben Becker, Gianlorenzo D'angelo, Esmaeil Delfaraz, and Hugo Gilbert. *Unveiling the Truth in Liquid Democracy with Misinformed Voters*. In: *Proceedings of the International Conference on Algorithmic Decision Theory (ADT)*. 2021, pp. 132–146. [↝ pp. 134, 136.]

[BDG+22] Markus Brill, Théo Delemazure, Anne-Marie George, Martin Lackner, and Ulrike Schmidt-Kraepelin. *Liquid Democracy with Ranked Delegations*. In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 2022, pp. 4884–4891. [↝ pp. 106, 108, 112, 121, 122, 130.]

[Beh17] Jan Behrens. *The Origins of Liquid Democracy*. In: *The Liquid Democracy Journal* 5 (2017). [↝ p. 108.]

[BF10] Steven Brams and Peter Fishburn. *Going from Theory to Practice: the Mixed Success of Approval Voting*. In: *Handbook on Approval Voting*. Springer, 2010, pp. 19–37. [↝ p. 20.]

[BF78] Steven Brams and Peter Fishburn. *Approval Voting*. In: *American Political Science Review* 72.3 (1978), pp. 831–847. [↝ p. 20.]

[BF82] Steven Brams and Peter Fishburn. *Deducing Preferences and Choices in the 1980 Presidential Election*. In: *Electoral Studies* 1.3 (1982), pp. 333–346. [↝ p. 20.]

[BFL+23] Markus Brill, Stefan Forster, Martin Lackner, Jan Maly, and Jannik Peters. *Proportionality in Approval-Based Participatory Budgeting*. In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 2023, pp. 5524–5531. [↷ p. 62.]

[BFN+19] Robert Bredereck, Piotr Faliszewski, Rolf Niedermeier, and Andrzej Kaczmarczyk. *An Experimental View on Committees Providing Justified Representation*. In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. 2019, pp. 109–115. [↷ p. 61.]

[BFN+20] Robert Bredereck, Piotr Faliszewski, Rolf Niedermeier, Piotr Skowron, and Nimrod Talmon. *Mixed Integer Programming with Convex/Concave Constraints: Fixed-Parameter Tractability and Applications to Multicovering and Voting*. In: *Theoretical Computer Science* 814 (2020), pp. 86–105. [↷ p. 43.]

[BGL19] Daan Bloembergen, Davide Grossi, and Martin Lackner. *On Rational Delegations in Liquid Democracy*. In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 2019, pp. 1796–1803. [↷ pp. 85, 109.]

[BHN+20] Matthias Bentert, Anne-Sophie Himmel, André Nichterlein, and Rolf Niedermeier. *Efficient Computation of Optimal Temporal Walks Under Waiting-Time Constraints*. In: *Applied Network Science* 5.1 (2020), pp. 1–26. [↷ pp. 113, 124–126.]

[BK15] Steven Brams and Marc Kilgour. *Satisfaction Approval Voting*. In: *Mathematical and Computational Modeling*. John Wiley and Sons, 2015. Chap. 11, pp. 273–298. [↷ p. 20.]

[BKN+14] Jan Behrens, Axel Kistner, Andreas Nitsche, and Björn Swierczek. *The Principles of LiquidFeedback*. Interacktive Demokratie, 2014. [↷ p. 105.]

[BKN+22] Jan Behrens, Axel Kistner, Andreas Nitsche, and Björn Swierczek. *The Temporal Dimension in the Analysis of Liquid Democracy Delegation Graphs*. In: *The Liquid Democracy Journal* 7 (2022). [↷ pp. 105, 108, 130.]

[BKS07] Steven Brams, Marc Kilgour, and Remzi Sanver. *A Minimax Procedure for Electing Committees*. In: *Public Choice* 3-4.132 (2007), pp. 401–420. [↷ pp. 20, 30.]

[BL16] Nathanaël Barrot and Jérôme Lang. *Conditional and Sequential Approval Voting on Combinatorial Domains*. In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. 2016, pp. 88–94. [↷ pp. 18–21, 24, 30–32, 38, 39, 59, 60, 62.]

[BMP+23] Markus Brill, Evangelos Markakis, Georgios Papasotiropoulos, and Jannik Peters. *Proportionality Guarantees in Elections with Interdependent Issues*. In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. 2023, pp. 2537–2545. [↷ pp. 13, 38.]

[Boc71] Frederick Bock. *An Algorithm to Construct a MINIMUM DIRECTED SPANNING TREE in a Directed Network*. In: *Developments in Operations Research* (1971), pp. 29–44. [↷ p. 121.]

[Bod96] Hans Bodlaender. *A Linear-Time Algorithm for Finding Tree-Decompositions of Small Treewidth*. In: *SIAM Journal On Computing* 25.6 (1996), pp. 1305–1317. [↷ p. 96.]

[BP03] Cristina Bazgan and Vangelis Paschos. *Differential Approximation for Optimal Satisfiability and Related Problems*. In: *European Journal of Operational Research* 147.2 (2003), pp. 397–404. [↷ p. 31.]

[Bri18] Markus Brill. *Interactive Democracy*. In: *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 2018, pp. 1183–1187. [↷ p. 108.]

[BS15] Jan Behrens and Björn Swierczek. *Preferential Delegation and the Problem of Negative Voting Weight*. In: *The Liquid Democracy Journal* 3 (2015). [↷ p. 122.]

[BTT92] John Bartholdi III, Craig Tovey, and Michael Trick. *How Hard is it to Control an Election?* In: *Mathematical and Computer Modelling* 16.8-9 (1992), pp. 27–40. [↷ pp. 40, 41.]

[BU09] Nadja Betzler and Johannes Uhlmann. *Parameterized Complexity of Candidate Control in Elections and Related Digraph Problems*. In: *Theoretical Computer Science* 410.52 (2009), pp. 5425–5442. [↷ pp. 53, 54, 56.]

[BZ16] Christian Blum and Christina Isabel Zuber. *Liquid Democracy: Potentials, Problems, and Perspectives*. In: *Journal of Political Philosophy* 24.2 (2016), pp. 162–182. [↷ pp. 83, 105, 130.]

[Cab04] Yves Cabannes. *Participatory Budgeting: A Significant Contribution to Participatory Democracy*. In: *Environment and Urbanization* 16.1 (2004), pp. 27–46. [↷ p. 134.]

[Car84] Lewis Carroll. *The Principles of Parliamentary Representation*. Harrison and Sons, 1884. [↷ p. 108.]

[CC08] Miroslav Chlebík and Janka Chlebíková. *Approximation Hardness of Dominating Set Problems in Bounded Degree Graphs*. In: *Information and Computation* 206.11 (2008), pp. 1264–1275. [↷ p. 90.]

[CCJ+05] David Cohen, Martin Cooper, Peter Jeavons, and Andrei Krokhin. *Supermodular Functions and the Complexity of MAX-CSP*. In: *Discrete Applied Mathematics* 149.1-3 (2005), pp. 53–72. [↷ p. 94.]

[CDG23] Rachael Colley, Théo Delemazure, and Hugo Gilbert. *Measuring a Priori Voting Power - Taking Delegations Seriously*. In: *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 2023, pp. 2658–2660. [↷ p. 109.]

[CEL+07] Yann Chevaleyre, Ulle Endriss, Jérôme Lang, and Nicolas Maudet. *A Short Introduction to Computational Social Choice*. In: *Proceedings of the International Conference on Current Trends in Theory and Practice of Computer Science*. 2007, pp. 51–69. [↷ p. 2.]

[CEL+08] Yann Chevaleyre, Ulle Endriss, Jérôme Lang, and Nicolas Maudet. *Preference Handling in Combinatorial Domains: From AI to Social Choice*. In: *AI Magazine* 29.4 (2008), pp. 37–37. [↷ p. 62.]

[CFK+15] Marek Cygan, Fedor Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. [↷ p. 23.]

[CFN+17] Jiehua Chen, Piotr Faliszewski, Rolf Niedermeier, and Nimrod Talmon. *Elections with Few Voters: Candidate Control can be Easy*. In: *Journal of Artificial Intelligence Research* 60 (2017), pp. 937–1002. [↷ p. 40.]

[CG17] Zoé Christoff and Davide Grossi. *Binary Voting with Delegable Proxy: An Analysis of Liquid Democracy*. In: *Proceedings of the Conference on Theoretical Aspects of Rationality and Knowledge (TARK)*. 2017, pp. 134–150. [↷ p. 85.]

[CGN21] Rachael Colley, Umberto Grandi, and Arianna Novaro. *Smart Voting*. In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. 2021, pp. 1734–1740. [↷ pp. 20, 85.]

[CGN22] Rachael Colley, Umberto Grandi, and Arianna Novaro. *Unravelling Multi-Agent Ranked Delegations*. In: *Journal of Autonomous Agents and Multi Agent Systems* 36.9 (2022), pp. 1–48. [↷ p. 121.]

[CHM+21] Arnaud Casteigts, Anne-Sophie Himmel, Hendrik Molter, and Philipp Zschoche. *Finding Temporal Paths Under Waiting Time Constraints*. In: *Algorithmica* 83.9 (2021), pp. 2754–2802. [↷ p. 122.]

[Chu65] Yoeng-Jin Chu. *On the Shortest Arborescence of a Directed Graph*. In: *Scientia Sinica* 14 (1965), pp. 1396–1400. [↷ p. 121.]

[CL12] Tugce Cuhadaroglu and Jean Lainé. *Pareto Efficiency in Multiple Referendum*. In: *Theory and Decision* 72.4 (2012), pp. 525–536. [↷ p. 20.]

[CM19] Ioannis Caragiannis and Evi Micha. *A Contribution to the Critique of Liquid Democracy*. In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. 2019, pp. 116–122. [↷ pp. 85, 124, 134.]

[CMM+17] Gal Cohensius, Shie Mannor, Reshef Meir, Eli A. Meirom, and Ariel Orda. *Proxy Voting for Better Outcomes*. In: *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 2017, pp. 858–866. [↷ pp. 108, 134, 136.]

[Cou90] Bruno Courcelle. *The Monadic Second-Order Logic of Graphs. I. Recognizable Sets of Finite Graphs*. In: *Information and Computation* 85.1 (1990), pp. 12–75. [↷ p. 95.]

[CQW+21] Yiwei Chen, Jingwen Qian, Junming Wang, Lirong Xia, and Gavriel Zahavi. *Opra: an Open-Source Online Preference Reporting and Aggregation System*. In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 2021, pp. 16011–16013. [↷ p. 8.]

[CS21] Alfonso Cevallos and Alistair Stewart. *A Verifiably Secure and Proportional Committee Election Rule*. In: *Proceedings of the ACM Conference on Advances in Financial Technologies (AFT)*. 2021, pp. 29–42. [↷ p. 134.]

[CW23] Andrei Constantinescu and Roger Wattenhofer. *Computing the Best Policy That Survives a Vote*. In: *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 2023, pp. 2058–2066. [↷ p. 138.]

[DGP98] Marc Demange, Pascal Grisoni, and Vangelis Paschos. *Differential Approximation Algorithms for Some Combinatorial Optimization Problems*. In: *Theoretical Computer Science* 209.1-2 (1998), pp. 107–122. [↷ p. 30.]

[DKN+11] Jessica Davies, George Katsirelos, Nina Narodytska, and Toby Walsh. *Complexity of and Algorithms for Borda Manipulation*. In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 2011, pp. 657–662. [↷ p. 41.]

[DL22] Hayrullah Dindar and Jean Lainé. *Compromise in Combinatorial Vote*. In: *Social Choice and Welfare* 59.1 (2022), pp. 1–32. [↷ p. 62.]

[DMS21] Palash Dey, Arnab Maiti, and Amatya Sharma. *On Parameterized Complexity of Liquid Democracy*. In: *Proceedings of the International Conference on Algorithms and Discrete Applied Mathematics (CALDAM)*. 2021, pp. 83–94. [↷ p. 109.]

[DPR+11] Giorgio Dalla Pozza, Maria Silvia Pini, Francesca Rossi, and K. Brent Venable. *Multi-Agent Soft Constraint Aggregation via Sequential Voting*. In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. 2011, pp. 172–177. [↷ p. 20.]

[DW71] Stuart Dreyfus and Robert Wagner. *The Steiner Problem in Graphs*. In: *Networks* 1.3 (1971), pp. 195–207. [↷ p. 130.]

[Edm67] Jack Edmonds. *Optimum Branchings*. In: *Journal of Research of the National Bureau of Standards - Mathematics and Mathematical Physics* 71B.4 (1967), pp. 233–240. [↷ p. 121.]

[EFL+15] Edith Elkind, Piotr Faliszewski, Martin Lackner, and Svetlana Obraztsova. *The Complexity of Recognizing Incomplete Single-Crossing Preferences*. In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 2015, pp. 865–871. [↷ p. 137.]

[EGP19] Bruno Escoffier, Hugo Gilbert, and Adèle Pass-Lanneau. *The Convergence of Iterative Delegations in Liquid Democracy in a Social Network*. In: *Proceedings of the International Symposium on Algorithmic Game Theory (SAGT)*. 2019, pp. 284–297. [↷ pp. 84, 85, 93, 97, 103, 108, 109.]

[EGP20] Bruno Escoffier, Hugo Gilbert, and Adèle Pass-Lanneau. *Iterative Delegations in Liquid Democracy with Restricted Preferences*. In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 2020, pp. 1926–1933. [↜ pp. 85, 97, 103.]

[End11] Ulle Endriss. *Computational Social Choice: Prospects and Challenges*. In: *Proceedings of the European Future Technologies Conference and Exhibition (FET)*. 2011, pp. 68–72. [↜ p. 2.]

[End13] Ulle Endriss. *Sincerity and Manipulation Under Approval Voting*. In: *Theory and Decision* 74 (2013), pp. 335–355. [↜ p. 18.]

[End17] Ulle Endriss. *Trends in Computational Social Choice*. Lulu. com, 2017. [↜ p. 2.]

[FHH+09] Piotr Faliszewski, Edith Hemaspaandra, Lane Hemaspaandra, and Jörg Rothe. *A Richer Understanding of the Complexity of Election Systems*. In: *Fundamental Problems in Computing: Essays in Honor of Professor Daniel J. Rosenkrantz*. Springer, 2009, pp. 375–406. [↜ p. 2.]

[FHH11] Piotr Faliszewski, Edith Hemaspaandra, and Lane Hemaspaandra. *Multimode Control Attacks on Elections*. In: *Journal of Artificial Intelligence Research* 40 (2011), pp. 305–351. [↜ p. 40.]

[FKP21] Rupert Freeman, Anson Kahng, and David Pennock. *Proportionality in Approval-Based Elections with a Variable Number of Winners*. In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. 2021, pp. 132–138. [↜ pp. 60, 62.]

[FL97] Moti Frances and Ami Litman. *On Covering Problems of Codes*. In: *Theory Computing Systems* 30.2 (1997), pp. 113–119. [↜ p. 148.]

[For02] Bryan Ford. *Delegative Democracy*. https://bford.info/deleg/deleg.pdf. Unpublished Manuscript. Accessed: 2023-09-26. 2002. [↜ p. 108.]

[For14] Bryan Ford. *Delegative Democracy Revisited*. https://bford.info/2014/11/16/deleg.html. Accessed: 2023-09-26. 2014. [↜ p. 108.]

[FR16] Piotr Faliszewski and Jörg Rothe. *Control and Bribery in Voting*. In: *Handbook of Computational Social Choice*. Cambridge University Press, 2016, pp. 146–268. [↜ p. 40.]

[Fre90] Eugene Freuder. *Complexity of k-Tree Structured Constraint Satisfaction Problems*. In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 1990, pp. 4–9. [↜ pp. 24, 25, 28.]

[FW22a] Robin Fritsch and Roger Wattenhofer. *The Price of Majority Support*. In: *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 2022, pp. 436–444. [↜ p. 62.]

[FW22b] Robin Fritsch and Roger Wattenhofer. *The Price of Majority Support*. In: *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 2022, pp. 436–444. [↜ p. 138.]

[FW92] Eugene Freuder and Richard Wallace. *Partial Constraint Satisfaction*. In: *Artificial Intelligence Journal* 58.1-3 (1992), pp. 21–70. [↜ p. 24.]

[GE10] Umberto Grandi and Ulle Endriss. *Lifting Rationality Assumptions in Binary Aggregation*. In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 2010, pp. 780–785. [↜ p. 20.]

[GKM+21] Paul Gölz, Anson Kahng, Simon Mackenzie, and Ariel D Procaccia. *The Fluid Mechanics of Liquid Democracy*. In: *ACM Transactions on Economics and Computation* 9.4 (2021), pp. 1–39. [↜ pp. 84, 85, 97, 106.]

[GKZ05] Avraham Goldstein, Petr Kolman, and Jie Zheng. *Minimum Common String Partition Problem: Hardness and Approximations*. In: *Electronic Journal of Combinatorics* 12 (2005). [↷ p. 32.]

[Gre15] James Green-Armytage. *Direct Voting and Proxy Voting*. In: *Constitutional Political Economy* 26.2 (2015), pp. 190–220. [↷ pp. 83, 108.]

[Gro07] Martin Grohe. *Logic, Graphs, and Algorithms*. In: *Electronic Colloquium of Computational Complexity* 91 (2007). [↷ p. 27.]

[Gro08] Martin Grohe. *Logic, Graphs, and Algorithms*. In: *Proceedings of the Logic and Automata: History and Perspectives [in Honor of Wolfgang Thomas]*. Vol. 2. 2008, pp. 357–422. [↷ p. 95.]

[GRS+22] Sushmita Gupta, Sanjukta Roy, Saket Saurabh, and Meirav Zehavi. *Resolute Control: Forbidding Candidates from Winning an Election is Hard*. In: *Theoretical Computer Science* 915 (2022), pp. 74–89. [↷ p. 40.]

[GS08] Georg Gottlob and Stefan Szeider. *Fixed-Parameter Algorithms for Artificial Intelligence, Constraint Satisfaction and Database Problems*. In: *The Computer Journal* 51.3 (2008), pp. 303–325. [↷ p. 29.]

[GSS01] Martin Grohe, Thomas Schwentick, and Luc Segoufin. *When is the Evaluation of Conjunctive Queries Tractable?* In: *Proceedings of the Symposium on the Theory of Computing (STOC)*. 2001, pp. 657–666. [↷ pp. 27, 29.]

[GV08] Petr Golovach and Yngve Villanger. *Parameterized Complexity for Domination Problems on Degenerate Graphs*. In: *Proceedings of the International Workshop on Graph-Theoretic Concepts in Computer Science*. 2008, pp. 195–205. [↷ p. 94.]

[GVY96] Naveen Garg, Vijay Vazirani, and Mihalis Yannakakis. *Approximate Max-Flow Min-(Multi) Cut Theorems and their Applications*. In: *SIAM Journal on Computing* 25.2 (1996), pp. 235–251. [↷ pp. 36, 37.]

[HFL15] Silu Huang, Ada Wai-Chee Fu, and Ruifeng Liu. *Minimum Spanning Trees in Temporal Graphs*. In: *Proceedings of the ACM SIGMOD* (2015), pp. 419–430. [↷ pp. 113, 118.]

[HHJ+23] Daniel Halpern, Joseph Halpern, Ali Jadbabaie, Elchanan Mossel, Ariel Procaccia, and Manon Revel. *In Defense of Liquid Democracy*. In: *Proceedings of the ACM Conference on Economics and Computation (EC)*. 2023, p. 852. [↷ pp. 134, 136.]

[HHR07] Edith Hemaspaandra, Lane Hemaspaandra, and Jörg Rothe. *Anyone but Him: The Complexity of Precluding an Alternative*. In: *Artificial Intelligence Journal* 171.5-6 (2007), pp. 255–285. [↷ pp. 40, 42, 44.]

[HKP+23] Daniel Halpern, Gregory Kehne, Ariel D Procaccia, Jamie Tucker-Foltz, and Manuel Wüthrich. *Representation with Incomplete Votes*. In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 2023, pp. 5657–5664. [↷ p. 134.]

[HL15] Steve Hardt and Lia Lopes. *Google Votes: A Liquid Democracy Experiment on a Corporate Social Network*. In: *Technical Disclosure Commons* (2015). [↷ p. 106.]

[HM15] Jac Heckelman and Nicholas Miller. *Handbook of Social Choice and Voting*. Edward Elgar Publishing, 2015. [↷ p. 2.]

[HP23] Wesley Holliday and Eric Pacuit. *Stable Voting*. In: *Constitutional Political Economy* (2023), pp. 1–13. [↷ p. 8.]

[IIB+22] Aviram Imber, Jonas Israel, Markus Brill, and Benny Kimelfeld. *Approval-Based Committee Voting under Incomplete Information*. In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 2022, pp. 5076–5083. [↷ p. 137.]

[Jan16] Svante Janson. *Phragmén's and Thiele's election methods*. In: *CoRR (arXiv:1611.08826)* (2016). [↷ p. 65.]

[JLR+17] Mark Jones, Daniel Lokshtanov, M. S. Ramanujan, Saket Saurabh, and Ondrej Suchý. *Parameterized Complexity of* DIRECTED STEINER TREE *on Sparse Graphs*. In: *SIAM Journal on Discrete Mathematics* 31.2 (2017), pp. 1294–1327. [↷ p. 130.]

[JST20] Pallavi Jain, Krzysztof Sornat, and Nimrod Talmon. *Participatory Budgeting with Project Interactions*. In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. 2020, pp. 386–392. [↷ p. 20.]

[JST22] Pallavi Jain, Krzysztof Sornat, and Nimrod Talmon. *Preserving Consistency for Liquid Knapsack Voting*. In: *Proceedings of the European Conference on Multi-Agent Systems (EUMAS)*. 2022, pp. 221–238. [↷ p. 109.]

[KHK02] Arie Koster, Stan van Hoesel, and Antoon Kolen. *Solving Partial Constraint Satisfaction Problems with Tree Decomposition*. In: *Networks* 40.3 (2002), pp. 170–180. [↷ pp. 24, 25.]

[Kil10] Marc Kilgour. *Approval Balloting for Multi-Winner Elections*. In: *Handbook on Approval Voting*. Springer, 2010, pp. 105–124. [↷ p. 20.]

[KKK+11] Meir Kalech, Sarit Kraus, Gal A Kaminka, and Claudia V Goldman. *Practical Voting Rules with Partial Information*. In: *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 2011, pp. 151–182. [↷ p. 137.]

[KKK02] David Kempe, Jon M. Kleinberg, and Amit Kumar. *Connectivity and Inference Problems for Temporal Networks*. In: *Journal of Computer and System Sciences* 64.4 (2002), pp. 820–842. [↷ p. 113.]

[KKM+19] Dusan Knop, Martin Koutecký, Tomás Masarík, and Tomás Toufar. *Simplified Algorithmic Metatheorems Beyond MSO: Treewidth and Neighborhood Diversity*. In: *Logical Methods in Computer Science* 15.4 (2019), 12:1–12:32. [↷ pp. 25, 99–102.]

[KKM+20] Telikepalli Kavitha, Tamás Király, Jannik Matuschke, Ildikó Schlotter, and Ulrike Schmidt-Kraepelin. *Popular Branchings and their Dual Certificates*. In: *Proceedings of the International Conference on Integer Programming and Combinatorial Optimization*. 2020, pp. 223–237. [↷ p. 85.]

[KL22] Aggelos Kiayias and Philip Lazos. *SoK: Blockchain Governance*. In: *Proceedings of the ACM Conference on Advances in Financial Technologies (AFT)*. 2022, pp. 61–73. [↷ p. 134.]

[KMP21] Anson Kahng, Simon Mackenzie, and Ariel Procaccia. *Liquid Democracy: An Algorithmic Perspective*. In: *Journal of Artificial Intelligence Research* 70 (2021), pp. 1223–1252. [↷ pp. 85, 134.]

[Kos09] Vassilis Kostakos. *Temporal Graphs*. In: *Physica A: Statistical Mechanics and its Applications* 388.6 (2009), pp. 1007–1023. [↷ p. 113.]

[KPR+97] Philip Klein, Serge Plotkin, Satish Rao, and Eva Tardos. *Approximation Algorithms for* STEINER *and* DIRECTED MULTICUTS. In: *Journal of Algorithms* 22.2 (1997), pp. 241–269. [↷ pp. 36, 37.]

[KR20] Grammateia Kotsialou and Luke Riley. *Incentivising Participation in Liquid Democracy with Breadth-First Delegation*. In: *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 2020, pp. 638–644. [↷ pp. 106, 122.]

[Kre08] Stephan Kreutzer. *Algorithmic Meta-Theorems*. In: *Proceedings of the International Workshop on Parameterized and Exact Computation*. 2008, pp. 10–12. [↷ p. 95.]

[Lac14] Martin Lackner. *Incomplete Preferences in Single-Peaked Electorates*. In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 2014, pp. 742–748. [↷ p. 137.]

[Lan20] Jérôme Lang. *Collective Decision Making Under Incomplete Knowledge: Possible and Necessary Solutions*. In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. 2020, pp. 4885–4891. [↷ p. 136.]

[LFZ+09] Hong Liu, Haodi Feng, Daming Zhu, and Junfeng Luan. *Parameterized Computational Complexity of Control Problems in Voting Systems*. In: *Theoretical Computer Science* 410.27-29 (2009), pp. 2746–2753. [↷ p. 40.]

[LL06] Gilbert Laffond and Jean Lainé. *Single-Switch Preferences and the Ostrogorski Paradox*. In: *Mathematical Social Sciences* 52.1 (2006), pp. 49–66. [↷ p. 62.]

[LL09] Gilbert Laffond and Jean Lainé. *Condorcet Choice and the Ostrogorski Paradox*. In: *Social Choice and Welfare* 32.2 (2009), pp. 317–333. [↷ p. 20.]

[LMM07] Rob LeGrand, Evangelos Markakis, and Aranyak Mehta. *Some Results on Approximating the Minimax Solution in Approval Voting*. In: *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 2007, pp. 1185–1187. [↷ p. 148.]

[LMX18] Jérôme Lang, Jérôme Mengin, and Lirong Xia. *Voting on Multi-Issue Domains with Conditionally Lexicographic Preferences*. In: *Artificial Intelligence Journal* 265 (2018), pp. 18–44. [↷ p. 32.]

[LRK23] Martin Lackner, Peter Regner, and Benjamin Krenn. *Abcvoting: A Python Package for Approval-Based Multi-Winner Voting Rules*. In: *Journal of Open Source Software* 8.81 (2023), p. 4880. [↷ p. 8.]

[LS10] Jean-François Laslier and Remzi Sanver. *Handbook on Approval Voting*. Springer Science & Business Media, 2010. [↷ p. 20.]

[LS23] Martin Lackner and Piotr Skowron. *Multi-Winner Voting with Approval Preferences*. Springer Briefs in Intelligent Systems. Springer, 2023. [↷ pp. 2, 20, 60, 62.]

[LX09] Jérôme Lang and Lirong Xia. *Sequential Composition of Voting Rules in Multi-Issue Domains*. In: *Mathematical Social Sciences* 57.3 (2009), pp. 304–324. [↷ p. 20.]

[LX16] Jérôme Lang and Lirong Xia. *Voting in Combinatorial Domains*. In: *Handbook of Computational Social Choice*. Cambridge University Press, 2016, pp. 197–222. [↷ pp. 17, 62.]

[Mar10] Dániel Marx. *Can You Beat Treewidth?* In: *Theory of Computing* 6.1 (2010), pp. 85–112. [↷ pp. 25, 27.]

[Mic16] Othon Michail. *An Introduction to Temporal Graphs: An Algorithmic Perspective*. In: *Internet Mathematics* 12.4 (2016), pp. 239–280. [↷ p. 113.]

[Mil69] James Miller. *A Program for Direct and Proxy Voting in the Legislative Process*. In: *Public Choice* 7.1 (1969), pp. 107–113. [↷ p. 108.]

[MM12] Konstantin Makarychev and Yury Makarychev. *Approximation Algorithm for Non-Boolean max k-csp*. In: *Proceedings of the International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*. 2012, pp. 254–265. [↷ p. 94.]

[MM17] Konstantin Makarychev and Yury Makarychev. *Approximation Algorithms for csps*. In: *The Constraint Satisfaction Problem: Complexity and Approximability*. 2017, pp. 287–325. [↷ p. 94.]

[MMS19] George Mertzios, Othon Michail, and Paul Spirakis. *Temporal Network Optimization Subject to Connectivity Constraints*. In: *Algorithmica* 81.4 (2019), pp. 1416–1449. [↷ pp. 113, 124, 126.]

[MNT16] Pasin Manurangsi, Preetum Nakkiran, and Luca Trevisan. *Near-Optimal UGC-hardness of Approximating max k-csp_r*. In: *Proceedings of the International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX)*. 2016, 15:1–15:28. [↷ p. 94.]

[MP20] Evangelos Markakis and Georgios Papasotiropoulos. *Computational Aspects of Conditional Minisum Approval Voting in Elections with Interdependent Issues*. In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. 2020, pp. 304–310. [↷ pp. 12, 62.]

[MP21a] Evangelos Markakis and Georgios Papasotiropoulos. *An Approval-Based Model for Single-Step Liquid Democracy*. In: *Proceedings of the International Symposium on Algorithmic Game Theory (SAGT)*. 2021, pp. 360–375. [↷ pp. 13, 108.]

[MP21b] Evangelos Markakis and Georgios Papasotiropoulos. *Winner Determination and Strategic Control in Conditional Approval Voting*. In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. 2021, pp. 342–348. [↷ pp. 12, 62, 73.]

[MP23] Evangelos Markakis and Georgios Papasotiropoulos. *As Time Goes By: Adding a Temporal Dimension Towards Resolving Delegations in Liquid Democracy*. In: *CoRR (arXiv:2307.12898)* (2023). [↷ pp. 13, 14.]

[MPR+08] Reshef Meir, Ariel Procaccia, Jeffrey Rosenschein, and Aviv Zohar. *Complexity of Strategic Behavior in Multi-Winner Elections*. In: *Journal of Artificial Intelligence Research* 33 (2008), pp. 149–178. [↷ p. 40.]

[MPT22] Evangelos Markakis, Georgios Papasotiropoulos, and Artem Tsikiridis. *On Improved Interval Cover Mechanisms for Crowdsourcing Markets*. In: *Proceedings of the International Symposium on Algorithmic Game Theory (SAGT)*. 2022, pp. 94–112. [↷ p. 14.]

[MST21] Reshef Meir, Fedor Sandomirskiy, and Moshe Tennenholtz. *Representative Committees of Peers*. In: *Journal of Artificial Intelligence Research* 71 (2021), pp. 401–429. [↷ p. 136.]

[MT20] Tomáš Masařík and Tomáš Toufar. *Parameterized Complexity of Fair Deletion Problems*. In: *Discrete Applied Mathematics* 278 (2020), pp. 51–61. [↷ p. 99.]

[MW13] Nicholas Mattei and Toby Walsh. *Preflib: A Library for Preferences http://www.preflib.org*. In: *Proceedings of the International Conference on Algorithmic Decision Theory (ADT)*. 2013, pp. 259–270. [↷ p. 38.]

[Pau14] Alois Paulin. *Through Liquid Democracy to Sustainable Non-Bureaucratic Government*. In: *Proceedings of the International Conference for E-Democracy and Open Government*. 2014, pp. 205–217. [↷ p. 108.]

[Pau20] Alois Paulin. *An Overview of Ten Years of Liquid Democracy Research*. In: *Proceedings of the International Conference on Digital Government Research*. 2020, pp. 116–121. [↷ pp. 83, 108, 134.]

[PPS21] Dominik Peters, Grzegorz Pierczyński, and Piotr Skowron. *Proportional Participatory Budgeting with Additive Utilities*. In: *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*. 2021, pp. 12726–12737. [↷ pp. 60, 62.]

[PS20a] Dominik Peters and Piotr Skowron. *Proportionality and the Limits of Welfarism*. In: *Proceedings of the ACM Conference on Economics and Computation (EC)*. 2020, pp. 793–794. [↷ p. 65.]

[PS20b] Marcus Pivato and Arnold Soh. *Weighted Representative Democracy*. In: *Journal of Mathematical Economics* 88 (2020), pp. 52–63. [↷ p. 136.]

[Puk14] Friedrich Pukelsheim. *Proportional Representation: Apportionment Methods and Their Applications*. Springer, 2014. [↷ p. 60.]

[REH23] Simon Rey, Ulle Endriss, and Ronald de Haan. *A General Framework for Participatory Budgeting with Additional Constraints*. In: *Social Choice and Welfare* 1 (2023), pp. 1–37. [↷ p. 20.]

[RHB+22] Manon Revel, Daniel Halpern, Adam Berinsky, and Ali Jadbabaie. *Liquid Democracy in Practice: An Empirical Analysis of its Epistemic Performance*. In: *Proceedings of the ACM Conference on Equity and Access in Algorithms, Mechanisms, and Optimization*. 2022. [↷ pp. 109, 134, 136.]

[Rot15] Jörg Rothe. *Economics and Computation*. Springer, 2015. [↷ p. 2.]

[RS86] Neil Robertson and Paul Seymour. *Graph Minors. II. Algorithmic Aspects of Tree-Width*. In: *Journal of Algorithms* 7.3 (1986), pp. 309–322. [↷ p. 23.]

[say04a] “sayke”. *Liquid Democracy*. https://web.archive.org/web/20040726071737/twistedmatrix.com/wiki/python/LiquidDemocracy. Accessed: 2023-09-25. 2004. [↷ p. 108.]

[say04b] “sayke”. *The Liquid Democracy Voting System*. https://web.archive.org/web/20040518131300/twistedmatrix.com/wiki/python/LiquidDemocracyVotingSystem. Accessed: 2023-09-25. 2004. [↷ p. 108.]

[SEL+17] Luis Sánchez-Fernández, Edith Elkind, Martin Lackner, Noberto Fernández, Jesús Fisteus, Pablo Basanta Val, and Piotr Skowron. *Proportional Justified Representation*. In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 2017, pp. 670–676. [↷ p. 62.]

[Sey14] Paul Seymour. *The Origin of the Notion of Treewidth*. https://cstheory.stackexchange.com/q/27317. Accessed at 2021-05-16. 2014. [↷ p. 95.]

[SG22] Piotr Skowron and Adrian Górecki. *Proportional Public Decisions*. In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 2022, pp. 5191–5198. [↷ pp. 60–64, 66, 70, 77, 81.]

[Sko21] Piotr Skowron. *Proportionality Degree of Multiwinner Rules*. In: *Proceedings of the ACM Conference on Economics and Computation (EC)*. 2021, pp. 820–840. [↷ p. 62.]

[Sla97] Petr Slavık. *A Tight Analysis of the Greedy Algorithm for Set Cover*. In: *Journal of Algorithms* 25.2 (1997), pp. 237–254. [↷ p. 92.]

[SS10] Marko Samer and Stefan Szeider. *Constraint Satisfaction with Bounded Treewidth Revisited*. In: *Journal of Computer and System Sciences* 76.2 (2010), pp. 103–114. [↷ pp. 29, 30.]

[Sze11] Stefan Szeider. *Monadic Second Order Logic on Graphs with Local Cardinality Constraints*. In: *ACM Transactions on Computational Logic* 12.2 (2011), pp. 1–21. [↷ p. 99.]

[Tar77] Robert Endre Tarjan. *Finding Optimum Branchings*. In: *Networks* 7.1 (1977), pp. 25–35. [↷ p. 121.]

[Ter23] Zoi Terzopoulou. *Voting with Limited Energy: A Study of Plurality and Borda*. In: *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 2023, pp. 2085–2093. [↷ p. 137.]

[Thi95] Thorvald Nicolai Thiele. *Om Flerfoldsvalg*. In: *Oversigt over det Kongelige Danske Videnskabernes Selskabs Forhandlinger* (1895), pp. 415–441. [↷ p. 65.]

[TKO21] Zoi Terzopoulou, Alexander Karpov, and Svetlana Obraztsova. *Restricted Domains of Dichotomous Preferences with Possibly Incomplete Information*. In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 2021, pp. 5726–5733. [↷ p. 137.]

[Tul67] Gordon Tullock. *Proportional Representation.* In: *Toward a Mathematics of Politics* (1967), pp. 144–157. [↷ p. 108.]

[Vaz03] Vijay V. Vazirani. *Approximation algorithms*. Springer Science & Business Media, 2003. [↷ p. 30.]

[WCK+16] Huanhuan Wu, James Cheng, Yiping Ke, Silu Huang, Yuzhen Huang, and Hejun Wu. *Efficient Algorithms for Temporal Path Computation*. In: *IEEE Transactions on Knowledge and Data Engineering* 28.11 (2016). [↷ p. 113.]

[Web78] Robert Weber. *Comparison of Public Choice Systems*. In: *Cowles Foundation Discussion Papers* 732 (1978). [↷ p. 20.]

[WS11] David Williamson and David Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, 2011. [↷ p. 30.]

[XC12] Lirong Xia and Vincent Conitzer. *Approximating Common Voting Rules by Sequential Voting in Multi-Issue Domains*. In: *Proceedings of the International Symposium on Artificial Intelligence and Mathematics (ISAIM)*. 2012. [↷ p. 20.]

[Yan19] Yongjie Yang. *Complexity of Manipulating and Controlling Approval-Based Multiwinner Voting*. In: *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*. 2019, pp. 637–643. [↷ p. 40.]

[ZG21] Yuzhe Zhang and Davide Grossi. *Power in Liquid Democracy*. In: *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*. 2021, pp. 5822–5830. [↷ p. 85.]

[ZYG19] Aizhong Zhou, Yongjie Yang, and Jiong Guo. *Parameterized Complexity of Committee Elections with Dichotomous and Trichotomous Votes*. In: *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 2019, pp. 503–510. [↷ p. 137.]