Efficient Computation and Strategic Control in Conditional Approval Voting[1]

Evangelos Markakis and Georgios Papasotiropoulos

Abstract

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. The price we have to pay when we move to this higher level of expressiveness is that we end up with a computationally hard rule. Motivated by this, we first 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. Furthermore, we investigate the complexity of problems related to the strategic control of such 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.

1 Introduction

Over the years, the field of social choice theory has focused more and more on decision making over combinatorial domains. This involves either *multi-winner elections* (e.g. for the formation of a committee) or elections for a set of issues that need to be decided upon simultaneously, often referred to as *multiple referenda*. As an example of the latter, think of a local community that needs to decide on possible facilities or services to be established, based on current available budget.

In this work, we focus on approval voting as a means for collective decision making, which offers a simple and easy to use format for running elections on multiple issues with multiple alternatives each, by having each voter express an approval or disapproval separately for each alternative of each issue. There is already a range of voting rules that are based on approval ballots, including the classic Minisum solution as well as more recently introduced methods (see Related Work).

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 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 (which she evaluates as more important); a group of friends may want to go to a certain movie theater only if they decide to have dinner at a nearby location; 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 etc. It is rather obvious that voting separately for each issue cannot provide a good solution in any of these settings. Consequently, 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 [3]. They defined the notion of a

[1]The results presented here will also appear in [36].

conditional approval ballot 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 a higher level of expressiveness implies higher computational complexity. The Minisum solution is efficiently computable in the standard setting but its generalization, referred to as *Conditional Minisum* (or CMS in short), is NP-hard. Hence, it becomes natural to investigate whether the problem admits exact algorithms for special cases or approximation algorithms with provable guarantees. Furthermore, we know nothing about the complexity of strategic aspects of a CMS election, which has been a very prominent research agenda within computational social choice.

Contribution. We first study algorithmic aspects of the Conditional Minisum voting rule for approval voting with preferential dependencies. In Section 3, we focus on conditions that lead to exact polynomial time algorithms. 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 the global dependency graph of an instance (taking the union of dependencies by all voters). Restrictions on the structure of the global 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 and efficiency of computation. These results also establish a connection with a well studied class of Constraint Satisfaction Problems, which can be of independent interest.

Moving on, in Section 4, we initiate for CMS the study of some standard notions of election control. These problems concern the attempt by an external agent to enforce a certain outcome by adding or deleting either voters or alternatives in the election. 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. Our findings reveal that CMS is sufficiently resistant against such moves.

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, other rules have also been considered, such as the Minimax solution [9], Satisfaction Approval Voting [8], and families based on Weighted Averaging Aggregation [2]. For surveys on the desirable properties of approval voting, we refer to [7] and [27]. None of these rules however allow voters to express dependencies. The first work that exclusively took this direction for approval-based elections is by Barrot and Lang [3]. 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. Driven by the NP-hardness results of [3], algorithmic aspects were further studied in our work in [35], where certain tractable special cases were identified. Our work in the current paper provides a generalization to some of these results.

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 [31, 1, 15, 40], compact representation languages [6, 33, 21], or completion principles for partial preferences [30, 13]. An extended survey for voting in combinatorial domains can be found in [32]. See also [12] for an informative work on both the proposed solution concepts and their applications in AI.

Finally, in our work we also consider some versions of election control that fall 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 [18]. Indicatively, the study of such problems with adding or deleting voters or alternatives began with the paper of Bartholdi et al. [4] and some subsequent works are, among others, [25, 17, 34].

2 Formal Background

Let $I = \{I_1, \ldots, I_m\}$ 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 $D = D_1 \times D_2 \times \cdots \times D_m$ be the set of all possible outcomes. Let also $V = \{1, \ldots, n\}$ 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 [3]. 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 (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 $\{I_{j_1}, I_{j_2}, \ldots, I_{j_k}\} \subseteq I$ 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, when all alternatives in t have occurred and the outcome of I_j is r. Both cases of zero and positive in-degree for an issue can be unified in the following definition of conditional approval ballots.

Definition 1. A conditional approval ballot of a voter i over issues $I = \{I_1, \ldots, I_m\}$ with domains $D_1, \ldots, D_m$ 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 $\{t : \{d_j^1, d_j^2, \ldots, d_j^k\}\}$, instead of $\{t : d_j^1\}$, $\{t : d_j^2\}$, $\ldots$, $\{t : d_j^k\}$. 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 $\{\emptyset : d_j\}$. We note also that even though there is a similarity between CP-nets and conditional ballots, in [3] it is highlighted that they induce different semantics and are incomparable.

An important quantity for parameterizing families of instances is the maximum in-degree[2] of each graph G_i, namely $\Delta_i = \max_{j \in [m]}\{|N_i^-(I_j)|\}$. Given a voter i with conditional ballot B_i, we will denote by B_i^j the restriction of her ballot for issue I_j. Moreover, a *conditional approval voting profile* is given by a tuple $P = (I, V, B)$, where $B = (B_1, B_2, \ldots, B_n)$.

Definition 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 $(I, \bigcup_{i \in [n]} E_i)$.

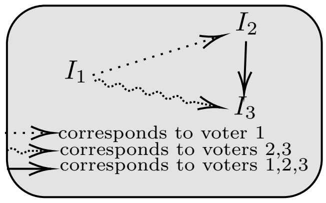
Example 1. As an illustration, we consider 3 co-authors of some joint research, several weeks before the submission deadline, who have to decide on 3 issues: whether they will *work* more before the submission on obtaining new theorems, whether they have enough

[2]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.

material to split their work into two, or even *multiple* papers or submit all their results in a single submission, and 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 $D_1 = \{w, \overline{w}\}$, $D_2 = \{m, \overline{m}\}, D_3 = \{c, \overline{c}\}$. The voters' preferences and the dependency graphs follow.

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



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 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 the projection of s on $N_i^-(I_j)$, say t, satisfies that $\{t : s_j\} \notin B_i^j$. We denote as $\delta_i(s)$ the total number of issues that dissatisfy voter i.

Coming back to Example 1, the values of $\delta_i(s)$ for every outcome s and voter i follow.

$\delta_i(\cdot)$	wmc	$wm\overline{c}$	$w\overline{m}c$	$w\overline{mc}$	$\overline{w}mc$	$\overline{w}m\overline{c}$	$\overline{wm}c$	$\overline{wmc}$
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 (which is $wm\overline{c}$ for the profile of Example 1). Formally, the algorithmic problem that we study is as follows.

CONDITIONAL MINISUM (CMS)	
Given:	A voting profile P with m binary issues and n voters casting conditional approval ballots.
Output:	A boolean assignment $s^* = (s_1^*, \dots, s_m^*)$ to all issues that achieves $\min_{s \in D} \sum_{i \in [n]} \delta_i(s)$.

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 [38] or to any textbook of parameterized complexity such as [14]. Note also that any missing proof is deferred to the Appendix.

3 Optimal Algorithms

The price we pay for the higher expressiveness of CMS is its increased complexity. Here, we focus on understanding the properties that allow CMS to be implemented in polynomial time. For this, we 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. To investigate what further restrictions can make the problem tractable, we utilize the global dependency graph of an instance, defined in Section 2, 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 [3], 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, 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.

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 [20] 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 4. The primal (or Gaifman) 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 then use known results from [19][3] or [29] for solving them efficiently.

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

[3] The original results in [19] do not deal with the optimization version, but as demonstrated in later works (e.g., Proposition 4.3 from [28]), it can be extended for this version.

Remark 1. Theorem 1 cannot be 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 (which is an essential part of the proof). On the other hand, it 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.

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 1, namely that binary CSPs can be reduced to solving CMS. Hence, together with Theorem 1, this means that CMS is computationally equivalent to binary CSP, and thus to any other problem for which the same result has been already established, e.g., such as the PARTITIONED SUBGRAPH ISOMORPHISM [37].

Theorem 2. *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. 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, x_k be the two variables involved in the 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 also satisfied unconditionally with every outcome for every other issue of the produced instance. Obviously, 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 $\mathcal{O}(d^2)$ 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 1. *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 allows us to apply some known hardness results on binary CSPs, namely [23, 24], 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 1 is essentially tight, and this resolves the problem of finding a characterization for instances that admit polynomial time solutions for CMS, subject to a standard complexity assumption.

Corollary 1. *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. If $\mathcal{G}$ is a class of graphs, as in the statement, then by Theorem 2, 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 now be completed by applying the hardness results for binary CSPs by [23, 24]. □

Remark 2. If we strengthen the complexity assumption used, from FPT $\neq$ W[1] to ETH, we can obtain an even stronger impossibility. In particular, by exploiting the result of [37], and the proof of Theorem 2, 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 $\mathcal{O}(n^{tw(G)})$ of the algorithm from Theorem 1 is the best possible up to an $\mathcal{O}(\log(tw(G)))$ factor in the exponent.

Parameterized complexity of CMS. The algorithm used in the proof of Theorem 1, runs in time exponential in $tw(G)$, where G is the global dependency graph and thus it places CMS in XP w.r.t 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 with binary CSP, we can use existing results [39, 22] 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 w.r.t. the parameter "treewidth + domain size". On the negative side, we cannot hope to prove FPT only w.r.t the treewidth parameter, independent of the domain size, as stated below.

Corollary 2. *When $\Delta_i \leq 1$ for every voter i,* CMS *is in FPT w.r.t the parameter $tw + d$, where tw is the treewidth of the global dependency graph and d is the maximum domain size. Moreover, it is $W[1]$-hard w.r.t. tw and w.r.t. d.*

4 Strategic Control of CMS Elections

In this section, we consider strategic aspects of CMS and study questions related to controlling an election of interdependent issues, which falls under the broad and well studied umbrella of influencing election outcomes. 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 [11] for related examples. 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. In this manner, we obtain 8 distinct algorithmic problems. Following the terminology of [25], 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, 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 [4]. As noted in [26], the "dream case" would be an efficiently computable voting rule which would be either resistant or immune to all control types. Hence, given the results of Section 3, we are mainly interested in elections that satisfy the conditions identified there, or even in further restricted cases. For an overview of the results we obtained in this section, we refer to the following Table.

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

Table 1: Results on Controlling CMS elections. R stands for Resistant, V for Vulnerable and I for Immune. For a CMS instance, we denote as Δ the maximum in-degree of every voter's dependency graph ($\Delta = \max_{i\in[n]} \Delta_i$), d the maximum domain size and n the number of voters.

4.1 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.

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 $p \in D$ 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 $(I, D, V \setminus V'', B)$)?

Remark 3. One has the option of either breaking ties in favor of the controller, if there are multiple optimal solutions in CMS (as in [16]), 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. [4]. 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).

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, resistance to control by adding or deleting voters can be established even for very simple forms of elections, without even the presence of conditional ballots, as in the next theorem.

Theorem 3. CAV-ALL *and* CDV-ALL *are NP-hard even for Unconditional Minisum and for binary domain in each issue.*

Theorem 3 may not be very surprising, since controlling all issues could be a quite strict requirement. The next step is to see whether such hardness results 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.

Proposition 1 (implied by [10])**.** 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 or conditional ballots. Indeed, it suffices to move to non-constant domain size, to establish NP-hardness.

Theorem 4. CAV-1 *and* CDV-1 *are NP-hard, even for Unconditional Minisum, but with non-constant domain size in at least one issue.*

We now have a complete picture for the unconditional setting and the hardness results transfer for conditional ballots too. The status of CDV-1 and CAV-1 for constant domain size in the presence of conditional ballots, remains as an open problem.

4.2 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, $|D_k \setminus D''| \geq 1$.

Remark 4. We firstly note that all the comments made in Remark 3 are applicable here as well. Also, we have not included CAA-ALL in our definitions as CMS is trivially immune to adding alternatives 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, either 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. 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.

Proposition 2. 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.

As soon as we move however to instances with conditional ballots, the problems do become hard (with the exception of Proposition 3). We start with the hardness of CDA-ALL.

Theorem 5. CDA-ALL *is 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.*

Moving to CDA-1 and CAA-1 we show that they behave similarly for non-constant domain. The proof of Theorem 6 below, shows a connection with some natural problems on graphs, that have been previously linked to election control for other voting rules [5].

Theorem 6. 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 in at least one issue.*

Proof. We will now only prove the hardness of CDA-1. The proof for CAA-1 is similar and the corresponding adjustments can be found in the Appendix. We will perform a reduction from the NP-hard problem MAX OUT-DEGREE DELETION (MOD) [5].

Instance: A directed graph $G = (V, E)$, a special vertex $p \in V$ and an integer $k \geq 1$.
Output: Does there exist $V' \subseteq V$ 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 $I = \{I_0, I_1, I_2, \ldots, I_n\}$. 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 outcome.

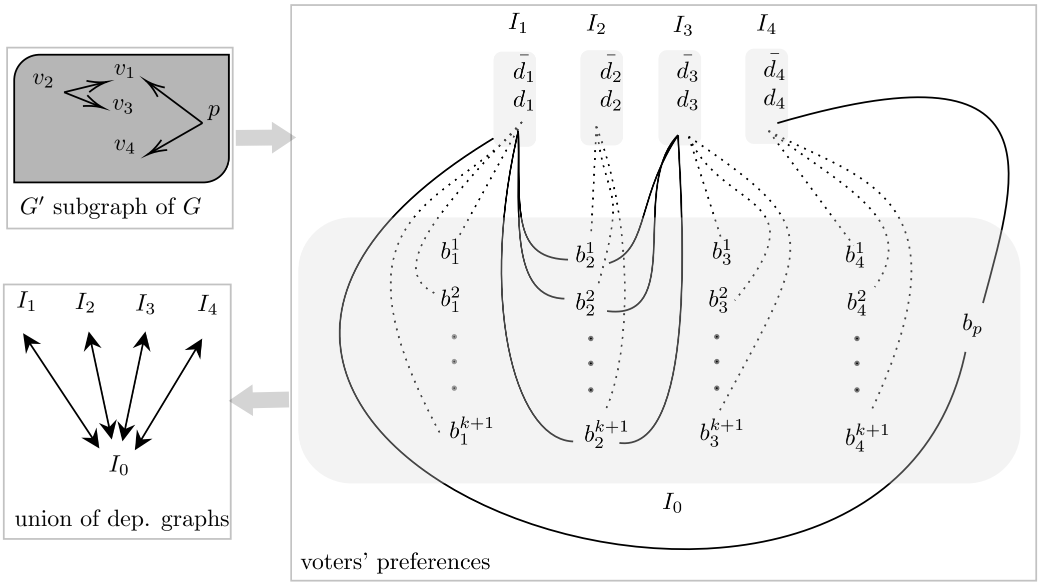
For every vertex other than p, we also have a block of L identical voters, where it suffices to take $L = m + 1$. Each voter in the j-th block, with $j \in V \setminus \{p\}$ has a dependency graph with 1 edge, from I_0 to I_j and votes as follows:

- 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 w.r.t. I_j only by d_j.
- For all other issues, she is satisfied with any outcome.

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 outcome 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 $\mathcal{O}(kn) = \mathcal{O}(n^2)$.

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

Figure 1: Illustrative example of the reduction in the proof of Theorem 6



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 Figure 1 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 WLOG 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 w.r.t. 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 w.r.t. 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 w.r.t. 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 WLOG 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 representative 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 $I_1, \ldots, I_n$ 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 w.r.t. 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. □

Moving to a constant domain size, CDA-1 and CAA-1 seem to behave differently.

Proposition 3. 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.

Hence, a constant domain size makes a difference for CDA-1 when we stick to the assumptions from Section 3 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). However, we have established intractability, as soon as we move to slightly richer instances with $\Delta_i \leq 2$.

Theorem 7. 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 binary domain size in every issue.*

Overall, we close current section concluding that CMS is indeed sufficiently computationally resistant in most of the variants of the control problem considered.

5 Conclusions and Further Work

We studied computational aspects of CMS elections, from the perspective of the winner determination problem using exact and parameterized algorithms as well as from the perspective of controlling election outcomes. We mostly focused on the case of $\Delta_i = 1$ as a non-trivial, natural, first-step generalization of the classic Minisum rule into conditional voting. For this case, we conclude that CMS provides a satisfactory tradeoff between expressiveness and efficiency under certain assumptions, and at the same time exhibits sufficient resistance to control in the considered settings.

There are still several interesting problems for future research. Algorithmic results for instances with even higher expressiveness (e.g., $\Delta_i \leq 2$) seem more challenging. Additionally, one can consider other objective functions, such as the Conditional Minimax rule, defined also in [3], for which, algorithmic results remain elusive. From a strategic point of view, some cases of our control questions have been left open, but more interestingly, one can go further and study other strategic moves such as destructive versions of control or bribery in a CMS election. Along this spirit, CMS was proven to be non-strategyproof by [3], but the complexity of finding the manipulation has not been examined yet.

References

[1] Stéphane Airiau, Ulle Endriss, Umberto Grandi, Daniele Porello, and Joel Uckelman. Aggregating dependency graphs into voting agendas in multi-issue elections. In *22nd International Joint Conference on Artificial Intelligence (IJCAI)*, 2011.

[2] Georgios Amanatidis, Nathanaël Barrot, Jérôme Lang, Evangelos Markakis, and Bernard Ries. Multiple referenda and multiwinner elections using Hamming distances: Complexity and manipulability. In *16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 715–723, 2015.

[3] Nathanaël Barrot and Jérôme Lang. Conditional and sequential approval voting on combinatorial domains. In *25th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 88–94, 2016.

[4] John J Bartholdi III, Craig A Tovey, and Michael A Trick. How hard is it to control an election? *Mathematical and Computer Modelling*, 16(8-9):27–40, 1992.

[5] Nadja Betzler and Johannes Uhlmann. Parameterized complexity of candidate control in elections and related digraph problems. *Theoretical Computer Science*, 410(52):5425–5442, 2009.

[6] Craig Boutilier, Ronen I. Brafman, Carmel Domshlak, Holger H. Hoos, and David Poole. CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *Journal of Artificial Intelligence Research*, 21:135–191, 2004.

[7] Steven J. Brams and Peter C. Fishburn. Going from theory to practice: the mixed success of approval voting. In *Handbook on Approval Voting*, pages 19–37. Springer, 2010.

[8] Steven J. Brams and D. Marc Kilgour. Satisfaction approval voting. Available at SSRN: https://ssrn.com/abstract=1608051, 2010.

[9] Steven J. Brams, D. Marc Kilgour, and M. Remzi Sanver. A minimax procedure for electing committees. *Public Choice*, 3-4(132):401–420, 2007.

[10] 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. *Theoretical Computer Science*, 2020.

[11] Jiehua Chen, Piotr Faliszewski, Rolf Niedermeier, and Nimrod Talmon. Elections with few voters: Candidate control can be easy. *Journal of Artificial Intelligence Research*, 60:937–1002, 2017.

[12] Yann Chevaleyre, Ulle Endriss, Jérôme Lang, and Nicolas Maudet. Preference handling in combinatorial domains: From AI to Social Choice. *AI Magazine*, 29(4):37–37, 2008.

[13] Tuğçe Çuhadaroğlu and Jean Lainé. Pareto efficiency in multiple referendum. *Theory and Decision*, 72(4):525–536, 2012.

[14] Marek Cygan, Fedor V Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized algorithms*, volume 4. Springer, 2015.

[15] Giorgio Dalla Pozza, Maria Silvia Pini, Francesca Rossi, and K. Brent Venable. Multi-agent soft constraint aggregation via sequential voting. In *22nd International Joint Conference on Artificial Intelligence (IJCAI)*, 2011.

[16] Jessica Davies, George Katsirelos, Nina Narodytska, and Toby Walsh. Complexity of and algorithms for borda manipulation. In *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, pages 657–662, 2011.

[17] Piotr Faliszewski, Edith Hemaspaandra, and Lane A Hemaspaandra. Multimode control attacks on elections. *Journal of Artificial Intelligence Research*, 40:305–351, 2011.

[18] Piotr Faliszewski and Jörg Rothe. Control and bribery in voting. In *Handbook of Computational Social Choice*, pages 146–268. Cambridge University Press, 2016.

[19] Eugene Freuder. Complexity of k-tree structured constraint satisfaction problems. In *Proceedings of the 4th AAAI Conference on Artificial Intelligence AAAI 1990*, pages 4–9, 1990.

[20] Eugene C Freuder and Richard J Wallace. Partial constraint satisfaction. *Artificial Intelligence*, 58(1-3):21–70, 1992.

[21] Christophe Gonzales, Patrice Perny, and Sergio Queiroz. GAI-networks: Optimization, ranking and collective choice in combinatorial domains. *Foundations of Computing and Decision Sciences*, 33(1):3–24, 2008.

[22] Georg Gottlob and Stefan Szeider. Fixed-parameter algorithms for artificial intelligence, constraint satisfaction and database problems. *The Computer Journal*, 51(3):303–325, 2008.

[23] Martin Grohe. The complexity of homomorphism and constraint satisfaction problems seen from the other side. *Journal of the ACM*, 54(1):1–24, 2007.

[24] Martin Grohe, Thomas Schwentick, and Luc Segoufin. When is the evaluation of conjunctive queries tractable? In *Proceedings of the 33rd Symposium on the Theory of Computing, STOC 2001*, pages 657–666, 2001.

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

[26] Edith Hemaspaandra, Lane A Hemaspaandra, and Jörg Rothe. Hybrid elections broaden complexity-theoretic resistance to control. *Mathematical Logic Quarterly*, 55(4):397–424, 2009.

[27] D. Marc Kilgour. Approval balloting for multi-winner elections. In *Handbook on Approval Voting*, pages 105–124. Springer, 2010.

[28] Dusan Knop, Martin Koutecký, Tomás Masarík, and Tomás Toufar. Simplified algorithmic metatheorems beyond MSO: treewidth and neighborhood diversity. *Logical Methods in Computer Science*, 15(4), 2019.

[29] Arie M. C. A. Koster, Stan P. M. van Hoesel, and Antoon W. J. Kolen. Solving partial constraint satisfaction problems with tree decomposition. *Networks*, 40(3):170–180, 2002.

[30] Gilbert Laffond and Jean Lainé. Condorcet choice and the Ostrogorski paradox. *Social Choice and Welfare*, 32(2):317–333, 2009.

[31] Jérôme Lang and Lirong Xia. Sequential composition of voting rules in multi-issue domains. *Mathematical Social Sciences*, 57(3):304–324, 2009.

[32] Jérôme Lang and Lirong Xia. Voting in combinatorial domains. In *Handbook of Computational Social Choice*, pages 197–222. Cambridge University Press, 2016.

[33] Minyi Li, Bao Quoc Vo, and Ryszard Kowalczyk. An efficient majority-rule-based approach for collective decision making with CP-nets. In *12th International Conference on the Principles of Knowledge Representation and Reasoning*, 2010.

[34] Hong Liu, Haodi Feng, Daming Zhu, and Junfeng Luan. Parameterized computational complexity of control problems in voting systems. *Theoretical Computer Science*, 410(27-29):2746–2753, 2009.

[35] Evangelos Markakis and Georgios Papasotiropoulos. Computational aspects of conditional minisum approval voting in elections with interdependent issues. In *29th International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 304–310, 2020.

[36] Evangelos Markakis and Georgios Papasotiropoulos. Winner determination and strategic control in conditional approval voting. In *30th International Joint Conference on Artificial Intelligence, IJCAI 2021, forthcoming*, 2021.

[37] Dániel Marx. Can you beat treewidth? *Theory of Computing*, 6:85–112, 2010.

[38] Neil Robertson and Paul D. Seymour. Graph minors. ii. algorithmic aspects of treewidth. *Journal of Algorithms*, 7(3):309–322, 1986.

[39] Marko Samer and Stefan Szeider. Constraint satisfaction with bounded treewidth revisited. *Journal of Computer and System Sciences*, 76(2):103–114, 2010.

[40] Lirong Xia and Vincent Conitzer. Approximating common voting rules by sequential voting in multi-issue domains. In *International Symposium on Artificial Intelligence and Mathematics (ISAIM)*, 2012.

A Missing Proofs from Section 3

Theorem 1. *If the global dependency graph of a* CMS *instance with* $\Delta_i \leq 1$ *for every voter* i*, has constant treewidth, then the problem is optimally solvable in polynomial time, even for 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 $k \in \mathcal{O}(1)$, 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 1, thus each constraint 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. Hence, the construction of the CSP instance can be done in polynomial time.

Since each constraint that involves two variables[4] 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. *The primal graph of the produced* CSP *instance is identical to the global dependency graph of the* CMS *instance.*

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 $\mathcal{O}(n^k)$ time by [19][5] or [29]. □

Corollary 2. *When* $\Delta_i \leq 1$ *for every voter* i*,* CMS *is in FPT w.r.t the parameter* $tw + d$*, where* tw *is the treewidth of the global dependency graph and* d *is the maximum domain size. Moreover, it is* $W[1]$*-hard w.r.t.* tw *and w.r.t.* d*.*

Proof. First, let us introduce some notation for ease of presentation. We will 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 $\mathcal{O}(f(S)|I|^c)$ for some constant c, and a computable function f, independent of any variable of Π other than the parameters in S. A set S dominates a set S' if whenever all parameters of S' are bounded by some constants, all parameters of S are bounded too. For a CMS instance having $\Delta_i \leq 1$ for every voter i, let tw be the treewidth of the global dependency graph, and let d be the maximum domain size over all issues. 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 arr the maximum number of variables that co-appear in a constraint.

To prove the positive statement one only needs to observe that CSP$\{arr, d', tw'\}$ is in FPT [24] and that Theorem 1 indicates that if $arr = 2$, then CMS$\{tw, d\}$ reduces to CSP$\{arr, d', tw'\}$. For the negative statement, due to Corollary 1, CMS$\{d\}$ cannot be in FPT unless FPT=W[1]. It remains to prove that even CMS$\{tw\}$ cannot be in FPT. In [39] (Theorem 1 therein), it was proved that binary CSP$\{S\}$ is not in FPT unless S dominates a limited number of parameter combinations. Using the domination lattice provided in the same work (Section 3 therein), we conclude that binary CSP$\{tw, arr\}$ is W[1]-hard, which due to Theorem 2, implies that CMS$\{tw\}$ is W[1]-hard too. □

[4]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.

[5]In fact, the original results in [19] do not deal with the optimization version, but as demonstrated in later works (see e.g., Proposition 4.3 from [28], it can be extended for this version as well.

B Missing Proofs from Section 4.1

Theorem 3. CAV-ALL *and* CDV-ALL *are NP-hard even for Unconditional Minisum and for binary domain in each issue.*

Proof. (i) We start with CDV-ALL, and 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 outcomes, 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 outcome $\overline{d_e}$ on 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 outcome $\overline{d_e}$ for every issue I_e is a 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 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 (it causes the same number of dissatisfactions as $\overline{d_e}$). This contradicts the fact that the removal of S resulted in a unique optimal solution with $\overline{d_e}$ selected for every issue I_e.

(ii) We continue now with the hardness of CAV-ALL. 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 [25]. For the sake of completeness, we provide the full construction here. 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 [25] (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.

We start with an instance P of EXACT-3-COVER (X3C) where $B = \{b_1, \ldots, b_m\}$ with $m = 3k$ is the universe, and $\mathcal{F} = \{S_1, \ldots, S_n\}$ is a collection of sets with $|S_i| = 3$, for every set S_i. 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.

We now 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 outcomes.
- 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 outcomes of S_i, and with w.

To finish the description, we set the quota parameter q equal to k and the desirable outcome of the controller to be $(\overline{b_1}, \dots, \overline{b_m}, w)$. 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 outcome 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 $(\overline{b_1}, \dots, \overline{b_m}, w)$.

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 outcomes, therefore, each b_j for $j \in [m]$, receives support by exactly one of the added voters. □

Theorem 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 [25]. 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 his 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. □

C Missing Proofs from Section 4.2

Proposition 2. CDA-1 and CDA-ALL can be solved in polynomial time whereas CAA-1 is immune, for Unconditional Minisum, with arbitrary domain size.

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). □

Theorem 5. CDA-ALL *is 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. Let $P = (G = (V, E), k)$ be an instance of Vertex Cover with n vertices and m edges, and a bound k on the size of the cover. We will present a reduction to an instance of CDA-ALL. Let there be one issue for every vertex of G and an extra issue I_0, hence $I = \{I_0, I_1, I_2, \ldots, I_n\}$, where for $j \in [n]$, $D_j = \{d_j, \overline{d_j}\}$. The domain of I_0 differs and contains $(k+1)m+1$ alternatives and in particular, for every edge e_i, there are $k+1$ alternatives e_i^ℓ, for $\ell \in [k+1]$. Essentially, these are identical $k+1$ 'copies' encoding the selection of e_i, and play a significant role in the reverse direction of the reduction. Additionally, D_0 contains one extra alternative, denoted by e_0. As for the voters, there is one edge voter for every edge $e_i = \{u, v\}$ of G, voting as follows:

- For the issue I_0, she supports unconditionally only the alternatives $\{e_i^\ell\}, \forall \ell \in [k+1]$ (these alternatives have an identical role for voter i).
- Concerning the issue I_u, she votes conditionally on I_0 for $\{e_i^\ell : d_u\} \forall \ell \in [k+1]$, and similarly for issue I_v, she votes $\{e_i^\ell : d_v\} \forall \ell \in [k+1]$.
- For all other issues she is satisfied with any outcome.

Additionally there are 3 dummy voters who are satisfied only with e_0 for issue I_0 and only with $\overline{d_j}$ for each $I_j, j \in [n]$. To complete the reduction, we use k as the quota parameter, and suppose the controller wants to enforce the outcome $p = (\{\overline{d_j}\}_{j \in [n]}, e_0)$ by removing at most k alternatives. It is easy to observe that for every voter i, $\Delta_i \leq 1$ in her dependency graph, and the global dependency graph is a star centered on I_0, thus with treewidth equal to 1. The maximum domain cardinality is $\mathcal{O}(km) = O(nm)$, which can be quadratic w.r.t. the number of voters.

We note that in the instance of CDA-ALL that we have constructed, the designated outcome p has a total dissatisfaction score of exactly $3m$ but it is not the unique winner.

If there exists a vertex cover in G of size at most k which is formed say WLOG by $\{v_1, \ldots, v_k\} \subseteq V$, we choose to delete the corresponding alternatives $\{d_1, \ldots, d_k\}$ from the issues $\{I_1, \ldots, I_k\} \subseteq I$. Let us count the cost of any solution that selects any e_i^ℓ as the choice for issue I_0 for $\ell \in [k+1]$. This dissatisfies $m-1$ among the edge voters w.r.t. I_0, and these voters will also be dissatisfied with the two issues corresponding to their edge, regardless of the assignment on the n issues, hence a total score of $3(m-1)$ from them. The i-th voter is satisfied with I_0 but will be dissatisfied with at least one issue from $I_1, \ldots I_n$, due to the vertex cover property. The three dummy voters are also dissatisfied with I_0. This results in a total dissatisfaction score of at least $3m+1$. Therefore, selecting the designated outcome p is now the unique optimal solution with a total dissatisfaction score of $3m$ (each of the first m voters is dissatisfied with exactly 3 issues).

For the reverse direction, suppose that there is a set D'' of at most k alternatives, the deletion of which, forces p to be the unique optimal solution. It is WLOG to assume that D'' does not contain anything from D_0. To elaborate on this claim, suppose that p was not the unique optimal solution before the deletion of the k alternatives, which means that another optimal solution existed choosing some e_i^ℓ on I_0, for some $i \in [m]$ and $\ell \in [k+1]$. By deleting up to k alternatives from D_0, one cannot completely eliminate all the alternatives that correspond to e_i, and this is the reason we added $k+1$ such copies. Hence, after the deletion of these alternatives, p would still not be a unique optimal solution. Thus, all the deletions concern the issues $I_1, \dots, I_n$. Moreover, none of the deleted alternatives can be in the form $\overline{d_j}$ for some issue I_j (we cannot delete alternatives that belong to our designated outcome p). To summarize, the deleted alternatives must come from distinct issues among $I_1, \dots I_n$ and each of them equals d_j for some $j \in [n]$. It is now easy to observe that the vertices corresponding to these alternatives in D'' form a vertex cover of size at most k in G. If not, there is some edge, say $e_i = (u, v)$ that is not covered. In that case, if we select e_i^ℓ for I_0 for some arbitrary ℓ, and d_j for all issues I_j where d_j has not been deleted, we can see that we will obtain a solution with the same dissatisfaction score of $3m$, contradicting the fact that p was a unique optimal solution. □

Theorem 6. 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 in at least one issue.*

Proof. The proof of NP-hardness of CDA-1 can be found in the main body of the paper. 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 defined and proved NP-hard in [5].

Instance: 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 $k \geq 1$.
Output: Does there exist 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[6], $D_j = \{\overline{d_j}\}$, 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, 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. □

Proposition 3. CDA-1 can be solved in polynomial time, when $\Delta_i \leq 1$ for every voter i, when the treewidth of the global dependency graph is constant and the domain size is also constant.

[6] 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.

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 $q = \mathcal{O}(1)$. But then we can check all possible ways of picking up to q items from the available set of all alternatives of all issues (a polynomial in m). For every such combination, and since the conditions of Theorem 1 hold, we can solve the remaining CMS instance and check if we can have the controller's choice in every optimal solution (by solving CMS with and without the designated alternative). □

Theorem 7. 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 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 $I = \{I_0, I_1, \ldots, I_n\}$. 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 w.r.t. 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 w.r.t. 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, we have ensured that 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, which means that 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. □

Evangelos Markakis
Athens University of Economics and Business
Athens, Greece
Email: markakis@gmail.com

Georgios Papasotiropoulos
Athens University of Economics and Business
Athens, Greece
Email: gpapasotiropoulos@gmail.com