

Winner Determination and Strategic Control in Conditional Approval Voting

Evangelos Markakis and Georgios Papasotiropoulos

Athens University of Economics and Business

{markakis, gpapasotiropoulos}@aueb.gr

Abstract

Our work focuses 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. Although the CMS rule provides much higher levels of expressiveness, this comes at the expense of increased computational complexity. In this work, we study further the issue of efficient algorithms for CMS, and we identify the condition of bounded treewidth (of an appropriate graph that emerges from the provided ballots), as the necessary and sufficient condition for polynomial algorithms, under common complexity assumptions. Additionally we investigate the complexity of problems related to the strategic control of such elections by the possibility of adding or deleting either voters or alternatives. We exhibit that in most variants of these problems, CMS is resistant against control.

1 Introduction

This work focuses on a voting rule for multi-issue elections that couples approval voting with the possibility of expressing dependencies among issues. As a first example, imagine a group of friends who have to decide on the menu of a buffet for the party they are hosting. It is expected that their preferences on certain dishes are conditioned on whether some other (e.g., complementary) dishes will be selected or not. We can 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 Alice’s satisfaction (the probability of an ad to be clicked by her). 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). The voters have conditional preferences, as the clicking probability is affected when an ad appears in a nearby slot with a related one (e.g. it is increased for frequently bought together products).

It seems obvious in the examples above, that deciding independently for each issue would not be a wise choice. Motivated by these considerations, the work of [Barrot and Lang, 2016] introduced an expressive framework that handles conditional ballots within the approval voting format, and defined, among others, the Conditional Minisum (CMS) voting rule. CMS can be thought of as a natural, intuitive generalization of the well known Minisum rule in approval-based elections, under preferential dependencies.

At the moment however, the power and limitations of CMS are still under-explored, and further studies are needed to arrive at a better evaluation of this relatively new rule. First, it has been shown that the higher level of expressiveness comes at the price of increased computational complexity; although the Minisum rule is easy to implement in the standard (unconditional) setting, CMS was proved to be intractable already by [Barrot and Lang, 2016]. Given this negative result, it becomes important to understand which properties can play a crucial role on obtaining efficient algorithms. The subsequent work of [Markakis and Papasotiropoulos, 2020] made some progress along this front, providing a sufficient condition for fast algorithms, but this has been far from a complete understanding. Second, we still know very little about the complexity of controlling and/or manipulating a CMS election. This is a very prominent research agenda within computational social choice, and further results would provide more insights on the benefits or drawbacks of conditional voting.

Contribution. Our main goal is to investigate algorithmic and complexity aspects of solving and controlling elections under CMS. More precisely, our results in Section 3, provide characterizations for the families of CMS instances that can be placed in P and FPT. The main insight gained out of this, is that the condition of bounded treewidth (of an appropriate graph, aggregating all the provided dependencies) serves as the lynchpin between expressiveness of preferences and efficiency of computation. These results also establish a connection between winner determination under CMS and 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. Our findings reveal that CMS is sufficiently resistant against such moves.

Related Work

The works most related to ours are [Barrot and Lang, 2016] and [Markakis and Papatotopoulos, 2020]. The former introduced the rule we study, and provided computational hardness results, along with other interesting properties, whereas the latter focused on algorithms for special cases of the problem. For surveys on approval voting, we refer to [Brams and Fishburn, 2010] and [Kilgour, 2010]. Beyond approval voting, we point to the work of [Chevaleyre *et al.*, 2008] for the relation of similar solution concepts within AI and to the enlightening survey of [Lang and Xia, 2016], for an exposition of solution concepts, proposed as attempts for a tradeoff between expressiveness and computational cost over combinatorial domains.

Regarding election control, the versions that we consider 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 [Faliszewski and Rothe, 2016]. Indicatively, the study of such problems with adding or deleting voters or alternatives began with the seminal paper of [Bartholdi III *et al.*, 1992] and some of the subsequent works are, among others, [Hemaspaandra *et al.*, 2007; Faliszewski *et al.*, 2011; Liu *et al.*, 2009].

2 Formal Background

For the relevant definitions of our setting, we closely follow the framework of [Barrot and Lang, 2016]. 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 different alternatives. Let $D = D_1 \times D_2 \times \dots \times D_m$ be the set of all feasible outcomes.

We consider a set $V = \{1, \dots, n\}$ of n voters who have to reach a common decision for every issue in I . The voters can express dependencies among issues in the following manner: we assume each voter i provides a directed graph $G_i = (I, E_i)$, referred to as her *dependency graph*. For an issue I_j , we denote by $N_i(I_j)$ the set of in-neighbors of the vertex corresponding to I_j in the dependency graph G_i . For an issue I_j where the set of in-neighbors is non-empty, the opinion of voter i on I_j is dependent on the outcome of the issues in $N_i(I_j)$. A useful quantity in the sequel is the maximum in-degree for a voter i , which is $\Delta_i = \max_{j \in [m]} \{|N_i(I_j)|\}$.

Definition 1. A conditional approval ballot of a voter i over the issues of $I = \{I_1, \dots, I_m\}$ 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 $\{t : d_j\}$, for some $t \in \prod_{k \in N_i(I_j)} D_k$ and some $d_j \in D_j$.

The rationale in Definition 1 is that the set A_j expresses the combinations of values that make the voter satisfied with respect to issue I_j . We note that when $N_i(I_j) = \emptyset$, voter i can express a simple unconditional ballot for approving an alternative d of issue I_j , which for simplicity, will be denoted as $\{d\}$. We also stress that our setting is incomparable with the semantics of the CP-nets framework, as already pointed out by [Barrot and Lang, 2016].

Definition 2. The global dependency graph of a set of voters is the undirected¹ simple graph that emerges from ignoring

the orientation of edges in the graph $(I, \bigcup_{i \in [n]} E_i)$.

Let $B = (B_1, \dots, B_n)$ denote the voters' conditional ballots. An instance of a conditional approval election is called a *profile*, given by a tuple (I, D, V, B) . An *outcome* of the election is a vector $(s_1, s_2, \dots, s_m) \in D$ (with $s_j \in D_j$).

Definition 3. Given an outcome $s = (s_1, s_2, \dots, s_m)$, we say that voter i is dissatisfied (or disagrees) with issue I_j , if either $N_i(I_j) = \emptyset$ and $\{s_j\} \notin B_i^j$, or if the projection of s on $N_i(I_j)$, say t , satisfies $\{t : s_j\} \notin B_i^j$, where B_i^j is the restriction of the ballot B_i on the issue I_j . We denote as $\delta_i(s)$ the total number of issues from s that dissatisfy voter i .

For illustrative examples concerning the definitions, we refer to [Barrot and Lang, 2016] and [Markakis and Papatotopoulos, 2020]. We focus on the following voting rule within the framework of conditional approval elections.

Problem: CONDITIONAL MINISUM (CMS)

Instance: A profile of a CMS election (I, D, V, B) .

Output: An outcome 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., $\Delta_i = 0, \forall i \in [n]$, then the election degenerates to Unconditional Minisum which is simply the classic Minisum rule in approval voting over multiple independent issues. In the sequel, we will make extensive use of the treewidth parameter of a graph G , denoted as $tw(G)$. For the relevant definition, we refer to [Robertson and Seymour, 1986] or to any textbook of parameterized complexity such as [Cygan *et al.*, 2015]. Finally, we note that all missing proofs can be found in the full version of this work.

3 Winner Determination of CMS Elections

The price we pay for the higher expressiveness of CMS is its increased complexity, as already established by the work that introduced the rule [Barrot and Lang, 2016]. Here, we focus on understanding the properties that allow CMS to be implemented in polynomial time. For this, we stick to the non-trivial case where $\Delta_i \leq 1$ for every voter i , which is already NP-hard, and investigate what further restrictions can make the problem tractable. More specifically, we utilize the global dependency graph of an instance, defined in Section 2, aggregating all the dependencies of the voters into a single graph. The role of the global dependency graph was highlighted in [Markakis and Papatotopoulos, 2020], where an optimal algorithm was described when the treewidth is at most 2. In this section, we first provide a generalization of this result for any constant treewidth (Theorem 1), resolving one of the open questions by [Markakis and Papatotopoulos, 2020]. But more importantly, we also show that this is tight, and we cannot hope to go beyond constant treewidth (Theorem 2 and Corollary 1), under standard complexity assumptions.

In our results, we make extensive use of *Constraint Satisfaction Problems* (CSPs), a prominent class of problems as they can model numerous applications in AI. 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

¹Defined as a directed graph in [Markakis and Papatotopoulos,

subset of the variables, and is represented by all the combinations of variables that make it satisfiable. 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 [Freuder and Wallace, 1992] 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 [Freuder, 1990]² or [Koster *et al.*, 2002] 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.*

Remark 1. Theorem 1 does not seem to have any direct generalization 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). Tackling higher degrees is left as an open problem. On the other hand, Theorem 1 can be generalized when there is a weight for each voter so as to optimize the weighted sum of the dissatisfaction scores.

Theorem 1 shows that $\Delta_i \leq 1$ for each i , along with constant treewidth on the global dependency graph form a sufficient condition for polynomial time solvability. We argue that such conditions are not overly restrictive for election rules. Instances with $\Delta_i \leq 1$ can be thought of as a natural first-step generalization of the unconditional approval voting case, by adding at most one dependence per issue. Constant treewidth also provides an adequate degree of expressiveness and can be encountered when the aggregate dependencies are not too dense. Graphs with low treewidth can arise when there is a commonly accepted hierarchy between issues or when voters have a high level of agreement on the type of dependencies in combination with the fact that $\Delta_i \leq 1$ for each voter i .

A natural question is whether we can solve other classes of instances, containing graphs of non-constant treewidth, under the assumption of $\Delta_i \leq 1$ for each i , by focusing on other parameters of the problem. Quite surprisingly, it turns out that the constant treewidth is essentially the only property that can yield efficiency guarantees. To establish this claim, we first show a “reverse” direction to Theorem 1, that binary CSPs can be reduced to CMS. Hence, together with Theorem 1, this means that CMS is computationally equivalent to binary

²The original results in [Freuder, 1990] do not deal with the optimization version, but as demonstrated in later works (e.g., Proposition 4.3 from [Knop *et al.*, 2019]), can be extended for this version.

CSPs, and thus to any other problem that is already known to have this property, such as the PARTITIONED SUBGRAPH ISOMORPHISM problem [Marx, 2010]. This reveals further connections between CMS and other combinatorial problems.

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 decision version of CSP asking for a solution that satisfies every constraint. 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 voter, and two issues I_1, I_2 (if they have not been introduced already by other constraints), 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 $\{x_1 : x_2\}, \{\bar{x}_1 : x_2\}, \{x_1 : \bar{x}_2\}$. In addition, the voter approves unconditionally every outcome of any issue other than I_2 .

To complete the reduction, we consider the decision version of CMS where we ask for an assignment with no dissatisfactions, i.e., the instance P' has an affirmative solution only when all voters are satisfied with all the issues. Observe 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). The following is quite obvious.

Claim 1. *The primal graph of CSP instance P is identical to the global dependency graph of the CMS instance P' .*

It remains to see that there exists a solution to P that satisfies every constraint if and only if there exists a solution to P' that satisfies every voter, which can be easily verified. \square

Theorem 2 allows us to apply some known hardness results on binary CSPs with large domain size, namely [Grohe, 2007; Grohe *et al.*, 2001], 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 computational complexity assumption.

Corollary 1. *Let \mathcal{G} be a recursively enumerable (e.g., decidable) class of graphs, and let $\text{CMS}(\mathcal{G})$ 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 $\text{FPT} \neq \text{W}[1]$, there is a polynomial algorithm for $\text{CMS}(\mathcal{G})$ if and only if every graph in \mathcal{G} has constant treewidth.*

Remark 2. If we strengthen the complexity assumption used, from $\text{FPT} \neq \text{W}[1]$ to the Exponential Time Hypothesis (ETH), we can obtain an even stronger impossibility result. In particular, by exploiting the result of [Marx, 2010], and the reduction in the proof of Theorem 2, we can show that under ETH, one cannot even hope for an algorithm on $\text{CMS}(\mathcal{G})$ that runs in time $f(G) \cdot |P|^{o(\text{tw}(G))/\log(\text{tw}(G))}$, where $|P|$ is the size of the CMS instance and $G \in \mathcal{G}$. This implies that the $\mathcal{O}(n^{\text{tw}(G)})$ algorithm from Theorem 1 is best possible up to an $\mathcal{O}(\log(\text{tw}(G)))$ factor in the exponent.

Parameterized complexity of CMS. The algorithm used in the proof of Theorem 1, runs in time exponential in $\text{tw}(G)$, 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 CSPs, we can use known results [Samer and Szeider, 2010; Gottlob and Szeider, 2008], to extract some further characterizations and obtain an almost complete picture with respect to the most relevant parameters. On the positive side, we can show 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 $\text{tw} + d$, where tw is the treewidth of the global dependency graph and d is the maximum domain size. Moreover, it is $\text{W}[1]$ -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 in computational social choice. 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 respectively, by disabling some existing voters, which can be done for example by changing the criteria for eligibility of voters (e.g., age, permanent residence, or even more specialized criteria in committee elections). Furthermore, a different instrument for the controller is to add more choices for the issues under consideration or delete existing ones, towards enforcing her will. We refer to [Chen *et al.*, 2017] for related examples. It is reasonable to assume that the controller does not have unlimited power, and therefore, she is capable of adding/deleting only a certain number of voters/alternatives.

Each combination of control features gives rise to a different control type. In this manner, we obtain 8 distinct algo-

rithmic problems that we study in this work, the formal definitions of which are presented in the following subsections. These are adaptations to CMS elections, of the original definitions provided in [Bartholdi III *et al.*, 1992]. Following the terminology of [Hemaspaandra *et al.*, 2007], 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. As noted in [Hemaspaandra *et al.*, 2009], 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. For an overview of the results of this section, we refer to Table 1.

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 [Davies *et al.*, 2011]), 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 III *et al.*, 1992]. 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 (intuitively simpler) extremes.

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.

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 appears to 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

	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.

domain size for the designated issue. The reason is that this can be reduced to an FPT version of the SET MULTI-COVER.

Proposition 1 (implied by [Bredereck *et al.*, 2020]). 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, the results of which, transfer to the conditional case too. The status of CDV-1 and CAV-1 for constant domain size in the presence of conditional ballots, remains unresolved.

4.2 Controlling Alternatives

We now consider the analogous control problems, regarding the addition or deletion of alternatives, instead of voters.

Instance: A CMS election (I, D, V, B) , where $D = D_1 \times \dots \times D_m$, 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 $D'' \subseteq \sqcup_{k \in [m]} D'_k$ (resp. $D'' \subseteq \sqcup_{k \in [m]} D_k$), 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 $D'' \subseteq \sqcup_{k \in [m]} D_k$, 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 , $|D_k \setminus D''| \geq 1$.

In the definitions above, we use \sqcup to denote the disjoint union of sets, having in mind that we could consider alternatives of different domains as distinct, even if they correspond to the same values (e.g. in boolean domains).

Remark 4. We first note that we have not included CAA-ALL in the definitions as CMS is trivially immune to adding alternatives in order to enforce a qualified alternative in every issue. Concerning CAA-1, we assume that the voters may express an opinion about any outcome of every issue, whether it

concerns a qualified or a spoiler alternative. 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 $|D_k \setminus D''| \geq 1$, for CDA-1 and CDA-ALL, is to ensure that the controller cannot eliminate all the alternatives of an issue.

It turns out that the picture differs significantly from the problem of adding or removing voters.

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.

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.

Next, we move to CDA-1 and CAA-1. When we allow a large domain size, we show that they have a similar behavior, and they are both resistant. The proof of Theorem 6 below, shows a connection with some natural problems on directed graphs, that have been linked to election control in the past for different voting rules [Betzler and Uhlmann, 2009].

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 prove hardness for CDA-1 (the proof for CAA-1 is similar) using a reduction from the NP-hard MAX OUT-DEGREE DELETION (MOD) [Betzler and Uhlmann, 2009].

Instance: A directed graph $G = (V, E)$, a distinguished vertex $p \in V$ and an integer $k \geq 1$.

Problem: 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 $I = \{I_0, I_1, I_2, \dots, I_n\}$. For $j \in [n]$, the domain of issue I_j is binary in the form $D_j = \{d_j, \bar{d}_j\}$. 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 , $\Delta_i \leq 1$, and the global dependency graph is a star centered on I_0 . The maximum domain cardinality is $\mathcal{O}(kn) = \mathcal{O}(n^2)$.

Suppose there exists a set S of vertices in G of size at most k , say WLOG that $S = \{1, \dots, k\} \subseteq V$, whose deletion leaves p as the only vertex of maximum out-degree. We now choose to delete the corresponding alternatives $\{d_1, \dots, d_k\}$ from the issues $\{I_1, \dots, I_k\}$. 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. 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 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. For the second case, suppose $j \in S$, then 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. We claim that 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 one 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 which contradicts the fact that b_p was elected for I_0 in every optimal solution. \square

Let us now move to a constant domain size for CDA-1 and CAA-1. Here, the problems 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, 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 (but 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 end the current section, concluding that CMS is computationally resistant in most of the variants of the control problem considered.

5 Conclusions

Our work has mostly focused on the natural, first-step generalization of the classic Minisum rule into conditional voting, i.e., with $\Delta_i \leq 1$ for every voter i . For this case, we conclude that CMS provides a satisfactory tradeoff between expressiveness and efficiency under the assumption of bounded treewidth, 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., with $\Delta_i \leq 2$, seem more challenging (the problem remains NP-hard and we are not aware of any properties that could lead to optimal algorithms for special cases). From a strategic point of view, some questions have been left open for the models we considered, including also parameterized complexity aspects. More interestingly, one can go further and study other strategic moves such as destructive versions of control or bribery aspects in a CMS election. Along this spirit, CMS was proven to be non-strategyproof by [Barrot and Lang, 2016], but the complexity of finding a manipulation has not been examined. Lastly, an experimental evaluation of the CMS election rule would also be an interesting pursuit, complementary to our theoretical analysis.

Acknowledgements

This work has been supported by the Hellenic Foundation for Research and Innovation (H.F.R.I.) under the "First Call for H.F.R.I. Research Projects to support faculty members and researchers and the procurement of high-cost research equipment" grant (Project Number: HFRI-FM17-3512).

References

- [Barrot and Lang, 2016] Nathanaël Barrot and Jérôme Lang. Conditional and sequential approval voting on combinatorial domains. In *Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI-16)*, pages 88–94, 2016.
- [Bartholdi III *et al.*, 1992] John Bartholdi III, Craig Tovey, and Michael Trick. How hard is it to control an election? *Mathematical and Computer Modelling*, 16(8-9):27–40, 1992.
- [Betzler and Uhlmann, 2009] Nadja Betzler and Johannes Uhlmann. Parameterized complexity of candidate control in elections and related digraph problems. *Theoretical Computer Science*, 410(52):5425–5442, 2009.
- [Brams and Fishburn, 2010] 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.
- [Bredereck *et al.*, 2020] 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*, pages 86–105, 2020.
- [Chen *et al.*, 2017] 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.
- [Chevalyere *et al.*, 2008] Yann Chevalyere, Ulle Endriss, Jérôme Lang, and Nicolas Maudet. Preference handling in combinatorial domains: From AI to Social Choice. *AI Magazine*, 29(4):37–37, 2008.
- [Cygan *et al.*, 2015] Marek Cygan, Fedor Fomin, Łukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michał Pilipczuk, and Saket Saurabh. *Parameterized algorithms*. Springer, 2015.
- [Davies *et al.*, 2011] Jessica Davies, George Katsirelos, Nina Nardyska, and Toby Walsh. Complexity of and algorithms for borda manipulation. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI-11)*, pages 657–662, 2011.
- [Faliszewski and Rothe, 2016] Piotr Faliszewski and Jörg Rothe. Control and bribery in voting. In *Handbook of Computational Social Choice*, pages 146–268. Cambridge University Press, 2016.
- [Faliszewski *et al.*, 2011] Piotr Faliszewski, Edith Hemaspaandra, and Lane Hemaspaandra. Multimode control attacks on elections. *Journal of Artificial Intelligence Research*, 40:305–351, 2011.
- [Freuder and Wallace, 1992] Eugene Freuder and Richard Wallace. Partial constraint satisfaction. *Artificial Intelligence*, 58(1-3):21–70, 1992.
- [Freuder, 1990] Eugene Freuder. Complexity of k -tree structured constraint satisfaction problems. In *Proceedings of the 4th AAAI Conference on Artificial Intelligence (AAAI-90)*, pages 4–9, 1990.
- [Gottlob and Szeider, 2008] Georg Gottlob and Stefan Szeider. Fixed-parameter algorithms for artificial intelligence, constraint satisfaction and database problems. *The Computer Journal*, 51(3):303–325, 2008.
- [Grohe *et al.*, 2001] 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-01)*, pages 657–666, 2001.
- [Grohe, 2007] Martin Grohe. The complexity of homomorphism and constraint satisfaction problems seen from the other side. *Journal of the ACM*, 54(1):1–24, 2007.
- [Hemaspaandra *et al.*, 2007] Edith Hemaspaandra, Lane Hemaspaandra, and Jörg Rothe. Anyone but him: The complexity of precluding an alternative. *Artificial Intelligence*, 171(5-6):255–285, 2007.
- [Hemaspaandra *et al.*, 2009] Edith Hemaspaandra, Lane Hemaspaandra, and Jörg Rothe. Hybrid elections broaden complexity-theoretic resistance to control. *Mathematical Logic Quarterly*, 55(4):397–424, 2009.
- [Kilgour, 2010] Marc Kilgour. Approval balloting for multi-winner elections. In *Handbook on Approval Voting*, pages 105–124. Springer, 2010.
- [Knop *et al.*, 2019] Dusan Knop, Martin Koutecký, Tomáš Masarík, and Tomáš Toufar. Simplified algorithmic metatheorems beyond MSO: treewidth and neighborhood diversity. *Logical Methods in Computer Science*, 15(4):12:1–12:32, 2019.
- [Koster *et al.*, 2002] Arie Koster, Stan van Hoesel, and Antoon Kolen. Solving partial constraint satisfaction problems with tree decomposition. *Networks*, 40(3):170–180, 2002.
- [Lang and Xia, 2016] Jérôme Lang and Lirong Xia. Voting in combinatorial domains. In *Handbook of Computational Social Choice*, pages 197–222. Cambridge University Press, 2016.
- [Liu *et al.*, 2009] 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.
- [Markakis and Papatotiropoulos, 2020] Evangelos Markakis and Georgios Papatotiropoulos. Computational aspects of conditional minisum approval voting in elections with interdependent issues. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI-20)*, pages 304–310, 2020.
- [Marx, 2010] Dániel Marx. Can you beat treewidth? *Theory of Computing*, 6:85–112, 2010.
- [Robertson and Seymour, 1986] Neil Robertson and Paul Seymour. Graph minors. II. algorithmic aspects of tree-width. *Journal of Algorithms*, 7(3):309–322, 1986.
- [Samer and Szeider, 2010] Marko Samer and Stefan Szeider. Constraint satisfaction with bounded treewidth revisited. *Journal of Computer and System Sciences*, 76(2):103–114, 2010.