Computational Aspects of Conditional Minisum Approval Voting in Elections with Interdependent Issues

Evangelos Markakis and **Georgios Papasotiropoulos**
Athens University of Economics and Business, Department of Informatics
{markakis, gpapasotiropoulos}@aueb.gr

Abstract

Approval voting provides a simple, practical framework for multi-issue elections, and the most representative example among such election rules is the classic Minisum approval voting rule. We consider a generalization of Minisum, introduced by the work of Barrot and Lang [2016], referred to as *Conditional Minisum*, where voters are also allowed to express dependencies between issues. 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 focus on the computational aspects of Conditional Minisum, where progress has been rather scarce so far. We identify restrictions to every voter's dependencies, under which we provide the first multiplicative approximation algorithms for the problem. The restrictions involve upper bounds on the number of dependencies an issue can have on the others. At the same time, by additionally requiring certain structural properties for the union of dependencies cast by the whole electorate, we obtain optimal efficient algorithms for well-motivated special cases. Overall, our work provides a better understanding on the complexity implications introduced by conditional voting.

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* (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. Approval voting offers a simple and easy to use format for running elections on multiple issues with binary domains, by having each voter express an approval or disapproval separately for 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. 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.

It is rather obvious that voting separately for each issue cannot provide a good solution in these settings. Instead, voters should be allowed to express dependencies among issues. Consequently, several approaches have been suggested to take into account preferential dependencies, see e.g., [Lang and Xia, 2016]. 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.

Barrot and Lang [2016] introduced a framework for expressing dependencies, tailored for approval voting elections. In particular, the notion of a conditional approval ballot was defined and new voting rules were introduced, that generalized some of the known rules in the standard setting of approval voting. Among the properties that were studied, it was also exhibited that a higher level of expressiveness implies higher computational complexity. Namely, the Minisum solution is efficiently computable in the standard setting but its generalization, referred to as *Conditional Minisum*, was proved to be $\mathbf{NP}$-hard. Beyond $\mathbf{NP}$-hardness, computational properties were not the main focus of [Barrot and Lang, 2016], and therefore, it has remained open whether the problem admits approximation algorithms with favorable guarantees or even exact algorithms for special cases.

Contribution. We focus on algorithmic aspects of the Conditional Minisum voting rule for multi-issue elections with preferential dependencies. Our main goal is to enhance our understanding on the complexity implications due to conditional voting for a rule that is known to be efficiently computable in the absence of dependencies. In Section 3, we provide the first multiplicative approximation algorithms for the problem under the condition that for every voter, each issue can depend on at most a constant number of other issues. In a convenient graph-theoretic representation, this corresponds to voters with dependency graph of constant maximum in-degree. Our family of approximation algorithms achieves a ratio that degrades smoothly as the in-degree grows larger. We stress that the problem is **NP**-hard even when every issue depends on at most one other issue, and for this case, our result yields a 2.2074-approximation ratio. Interestingly, our algorithms are based on a reduction to MIN SAT, an optimization version of SAT that has rarely been applied in computational social choice (in contrast to MAX SAT). Moving on, in Section 4, we focus on special cases that are optimally solvable in polynomial time. For this we stick to the (hard) case of maximum in-degree one. Our main insight is that one can draw conclusions by looking at the *global* dependency graph (taking the union of dependencies by all voters). Restrictions on the structure of the global graph allows us to identify several cases (e.g., trees, cycles, and more generally graphs with treewidth at most 2), where we can have optimal efficient algorithms. Hence we conclude with a positive confirmation that Conditional Minisum can combine enhanced expressiveness with efficient computation (for exact or approximate solutions) in many cases of well-motivated scenarios.

Related work. Apart from the classic Minisum solution, many other approval voting rules have been considered, such as the Minimax solution [Brams *et al.*, 2007], Satisfaction Approval Voting [Brams and Kilgour, 2010], and other families based on Weighted Averaging aggregation [Amanatidis *et al.*, 2015]. For surveys on the desirable properties of approval voting, we refer to [Brams and Fishburn, 2010] and [Kilgour, 2010]. None of these rules however allow voters to express dependencies. The first work that exclusively took this direction and is most closely related to ours is [Barrot and Lang, 2016]. Namely, three voting rules were proposed for incorporating such dependencies (including the Conditional Minisum rule that we consider here) and some of their properties were studied mainly on the satisfiability of certain axioms. 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 [Lang and Xia, 2009], [Airiau *et al.*, 2011], [Dalla Pozza *et al.*, 2011], [Xia and Conitzer, 2012], compact representation languages [Boutilier *et al.*, 2004], [Li *et al.*, 2010], [Gonzales *et al.*, 2008], or completion principles for partial preferences [Laffond and Lainé, 2009], [Çuhadaroğlu and Lainé, 2012]. An extended survey for voting in combinatorial domains can be found at [Lang and Xia, 2016]. See also [Chevaleyre *et al.*, 2008] for an informative work on both the proposed solution concepts and their applications in AI.

2 Formal Background

Let $I = \{I_1, \ldots, I_m\}$ be a set of m issues, each of them associated with a finite domain D_i. We only examine the case of binary domains so that for every $i \in [m]$, $D_i = \{d_i, \overline{d_i}\}$. Here d_i depicts a ballot in favor of the issue, whereas $\overline{d_i}$ is against it. 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 in D.

Voting format. To express dependencies among issues, we mostly follow the format described in [Barrot and Lang, 2016]. 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 explain briefly how voters submit their preferences, before giving the formal definition. For an issue I_j with no predecessors in G_i (its in-degree is 0), voter i is allowed to cast an unconditional approval ballot, stating the outcomes of I_j that are approved by her. She can be satisfied with one or with all or with none of the outcomes in D_j. In the case that issue I_j has a positive in-degree in G_i, 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 $\{t : d\}$, where $d \in D_j$, and $t \in D_{j_1} \times D_{j_2} \times \cdots \times D_{j_k}$. Every combination $\{t : d\}$ signifies the satisfaction of voter i with respect to issue I_j, when all outcomes implied by t have been realized and the outcome of I_j is d. Both cases of zero and positive in-degree for an issue can be unified in the following definition.

Definition 1. *A conditional approval ballot of a voter i over issues $I = \{I_1, \ldots, I_m\}$ with binary domains, 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 $\{t : d\}$, where $d \in D_j$, $t \in \prod_{k \in N_i^-(I_j)} D_k$, and $N_i^-(I_j)$ is the (possibly empty) set of direct predecessors of I_j in G_i.*

To simplify the presentation of a conditional ballot, when a voter has expressed a common dependency for the two outcomes of an issue, we can group them together and write $\{t : \{d_j, \overline{d_j}\}\}$, instead of $\{t : d_j\}$, $\{t : \overline{d_j}\}$. 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 $\{d_j\}$, since $N_i^-(I_j) = \emptyset$.

An important quantity for parameterizing families of instances in the sequel, is the maximum in-degree of each graph G_i. Namely, for a voter i let $\Delta_i = \max\{|N_i^-(I_j)|, j \in [m]\}$.

Given a voter i with conditional ballot B_i, we will denote by B_i^j the restriction (i.e., projection) of her ballot on issue I_j. Moreover, a *conditional approval voting profile*[1] P (often referred to simply as a *profile* for brevity) is a tuple (I, V, B) where $B = (B_1, B_2, \ldots, B_n)$.

Example 1. As an illustration, we consider 3 co-authors of some joint research, several weeks before a conference 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 material to split their work into two, or even *multiple*, papers and whether they

[1] When Δ_i is large for some voter i, the size of a profile might become exponential. Alternatively, one could aim for a succinct representation, e.g., via propositional formulas. We do not examine further this issue, since we consider instances with constant in-degree.

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, she has no strong opinion on multiple submissions and approves both alternatives, and she agrees with inviting a new co-author only if they decide both to work more for new results and to submit a single paper. The last author also expresses a dependence for inviting a new co-author on the other two issues, as described below.

More formally, let $I = \{I_1, I_2, I_3\}$ be the aforementioned issues, and for $i = 1, 2, 3$, let $G_i = (I, E_i)$ be the dependency graph of voter i, so that $E_1 = \{(I_1, I_2), (I_2, I_3)\}$, $E_2 = E_3 = \{(I_1, I_3), (I_2, I_3)\}$. Let also $D_1 = \{w, \overline{w}\}$, $D_2 = \{m, \overline{m}\}$, $D_3 = \{c, \overline{c}\}$. The voters' preferences are:

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

To measure the dissatisfaction of a voter given an assignment of values to all the issues, we use the following generalization of Hamming distance.

Definition 2. *Given an outcome $s = (s_1, s_2, \ldots, s_m) \in D$, we say that voter i is dissatisfied with issue I_j, if the projection of s on $N^-_i(I_j)$, say t, satisfies $\{t : s_j\} \notin B^j_i$. We denote as $\delta_i(s)$ the total number of issues that dissatisfy voter i.*

Coming back to Example 1, the values of $\delta_i(s)$ follow.

$\delta_i(\cdot)$	wmc	$wm\overline{c}$	$w\overline{m}c$	$w\overline{m}\overline{c}$	$\overline{w}mc$	$\overline{w}m\overline{c}$	$\overline{w}\overline{m}c$	$\overline{w}\overline{m}\overline{c}$
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

Finally, even though there is a similarity between CP-nets and conditional ballots, Barrot and Lang [2016] highlighted that they induce different semantics and are incomparable.

Voting rule. In this work, we study a generalization of the classic Minisum solution in the context of conditional approval voting. We refer to this rule as *Conditional Minisum* (CMS), and it outputs the outcome that minimizes the total number of dissatisfactions over all voters ($wm\overline{c}$ for the profile presented in Example 1). Formally, the algorithmic problem that our work deals with 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 $s^* = (s^*_1, \ldots, s^*_m)$ to all issues that achieves $\min_{s \in D} \sum_{i \in [n]} \delta_i(s)$.

3 Approximation Algorithms

It is well known that a Minisum solution can be efficiently computed when there are no dependencies [Brams *et al.*, 2007]. In contrast to this, CMS is $\mathbf{NP}$-hard even when there is only a single dependence per voter, i.e., when every voter's dependency graph has just a single edge [Barrot and Lang, 2016]. Given that hardness result, it is natural to resort to the framework of approximation algorithms. The only known result from this perspective is an algorithm with a *differential* approximation ratio of $4.34/(m \sum_{j \in I} 2^{|N^-(j)|} + 4.34)$ for the case of a common acyclic dependency graph, so that for each voter i and issue j, $N^-(j) = N^-_i(j)$ [Barrot and Lang, 2016]. However, differential approximations (we refer to [Demange *et al.*, 1998] for the definition) form a less typical approach in the field of approximation algorithms. Instead, we focus on the more standard framework of *multiplicative* approximation algorithms, as treated also in common textbooks [Vazirani, 2003], [Williamson and Shmoys, 2011]. We say that an algorithm for a minimization problem achieves a multiplicative ratio of $\alpha \geq 1$, if for every instance I, it produces a solution with cost at most α times the optimal. We stress that a differential approximation ratio for minimization problems does not in general imply any multiplicative approximation ratio [Bazgan and Paschos, 2003].

Our main contribution in this section is the first class of multiplicative approximation algorithms for CMS under the condition of bounded in-degree in every voter's dependency graph. To this end, we make use of approximation algorithms for the MIN k-SAT problem, a minimization version of SAT, where we are given a set of m clauses in k-CNF and we search for a boolean assignment so as to minimize the total number of satisfied clauses. Interestingly, minimization versions of SAT have rarely been applied in the context of computational social choice, see e.g., [Lang *et al.*, 2018]. The use of MAX SAT is much more common, but for the case of CMS, it does not seem convenient to exploit algorithms for maximisation versions of SAT, by following an analogous approach as in the proof of Theorem 1 below. We defer further discussion to the full version of our work.

We first present a result for profiles where $\Delta_i \leq 1$ for every voter i. This is already a superclass of the case that was proved $\mathbf{NP}$-hard in [Barrot and Lang, 2016], as we also allow for cycles. We later generalize to profiles of bounded Δ_i.

Theorem 1. *If the dependency graph of every voter has maximum in-degree at most 1, an α-approximation algorithm for* MIN 2-SAT *yields a 2α-approximation algorithm for* CMS*. In particular, we can have a polynomial time* 2.2074*-approximation for* CMS.

Proof. Consider an instance P of CMS, with n voters, and with the stated properties. We present a reduction to MIN 2-SAT that preserves the approximation up to a factor of 2.

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

For this we consider two cases. The first and easier case is when for a voter i, and issue I_j, $N_i^-(I_j) = \emptyset$. All possible forms of B_i^j are depicted in the first row of Table 1, whereas the corresponding formula is shown in the second row.

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

Table 1: The formula when issue I_j has no predecessor in G_i.

On the other hand, if I_j has an in-neighbor (it can have only one by our assumptions), say $I_k \in I$, we set C_{ij} equal to the disjunction of all combinations of outcomes on issues I_j and I_k that dissatisfy voter i with respect to I_j. To illustrate this construction, we describe an example with 4 voters, 2 issues $I = \{I_k, I_j\}$ and for every voter i, $G_i = \{I, \{I_k, I_j\}\}$. The preferences for issue I_j are shown in Table 2. Namely, for $i = 1, 2, 3, 4$, the first cell in the i-th row depicts B_i^j from which C_{ij} can be obtained as the disjunction of the ticked expressions in the remaining of the i-th row.

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

Table 2: For $i = 1, 2, 3, 4$ the formula C_{ij} is the disjunction of the ticked expressions in the i-th row.

Claim 1. *Considering an outcome $(s_1, \ldots, s_m)$ for the issues and the corresponding assignment to the variables, voter i is dissatisfied with I_j if and only if the formula C_{ij} is true.*

The constructed formula C_{ij} is in DNF. To continue, we will need to make a conversion to CNF, which is easy to do given its small size as per the following lemma. Its proof (based on a case analysis), along with some proofs of subsequent results have been omitted due to space constraints.

Lemma 1. *The formula C_{ij} for each voter $i \in V$, and each issue $I_j \in I$, can be written in CNF with at most 2 clauses, and where each clause contains at most 2 literals.*

Using Lemma 1 to convert each C_{ij} to CNF, we can now create a MIN 2-SAT instance P' by the multiset[2] of all clauses appearing in the C_{ij}'s, i.e., appearing in the formula

$$C = \bigwedge_{i \in V, I_j \in I} C_{ij}. \quad (1)$$

In the instance P', we aim for a truth assignment minimizing the number of satisfied clauses in C. Hence, our construction gives rise to the following algorithm for CMS.

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

Algorithm 1 ▷Input: P

1: Create P' from P using Lemma 1 and Equation (1).
2: Run an α-approximation of MIN 2-SAT on P'.
3: Set the value of I_j in P to the value of x_j in P'.

Lemma 2. *Let OPT(P) and OPT(P') be the values of the optimal solutions for the instances P and P' of* CMS *and* MIN 2-SAT *respectively. Then OPT$(P') \leq 2$OPT(P).*

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

$$\text{SOL}(P) \leq \text{SOL}(P') \leq \alpha \cdot \text{OPT}(P') \leq 2\alpha \cdot \text{OPT}(P)$$

Thus every α-approximation algorithm for MIN 2-SAT yields a 2α-approximation algorithm for CMS. Finally, to obtain the claimed approximation ratio of Theorem 1, we just need to use the algorithm by Avidor and Zwick [2005], which achieves a factor of 1.1037 for MIN 2-SAT. □

Suppose now that for every voter i, $\Delta_i \leq 2$. If we follow the approach in the proof of Theorem 1, it is simply a matter of boolean algebra to check that, in analogy to Lemma 1, we can write any resulting C_{ij} in CNF with at most 4 clauses, each containing at most 3 literals, for every $i \in V, I_j \in I$. We can then proceed, with a lemma similar to Lemma 2, and finally use the 1.2136-approximation algorithm for MIN 3-SAT [Avidor and Zwick, 2005] to obtain a ratio of 4.8544.

In fact the same approach can be further generalized, as long as the maximum in-degree in every voter's graph is bounded by a constant[3] k. In that case, the approach of Theorem 1 yields CNF formulas with $k + 1$ literals and at most 2^k clauses for each voter. Hence, by using the 2-approximation algorithm for MIN SAT by [Marathe and Ravi, 1996] (it applies to MIN k-SAT for any k), we have the following result.

Theorem 2. *If the dependency graph of every voter has maximum in-degree at most a constant k, there is a polynomial time 2^{k+1}-approximation algorithm for* CMS.

Finally, regarding tightness, we can show that Algorithm 1 cannot produce an approximation better than 2 (by creating instances that make Lemma 1 tight on the number of clauses). Therefore, a small gap between 2 and 2.2074 still remains. For the algorithms described in Theorem 2, by using similar constructions, we can show that their approximation ratio can be no better than 2^k when the in-degree is at most k. We defer further discussion for the full version of our work.

[3] For non-constant in-degree, the conversion from DNF to CNF in Lemma 1 may take exponential time.

4 Optimal Algorithms

In the current section, we identify special cases of the problem where exact optimal solutions can be found in polynomial time. In doing this, we stick to the assumption that every voter has maximum in-degree at most 1 in her graph. Since this already makes the problem $\mathbf{NP}$-hard, one needs to consider further restrictions that admit efficient algorithms. In our quest to define tractable cases, we realized that it is convenient to focus on the union of all the dependency graphs:

Definition 3. *The global dependency graph of a profile* $(I, [n], B)$ *is the graph* $(I, \bigcup_{i\in[n]} E_i)$*, i.e., we take the union of the edges in every voter's dependency graph.*

To see how to exploit the global dependency graph, it is instructive to inspect the $\mathbf{NP}$-hardness proof for CMS in [Barrot and Lang, 2016]. Their proof holds for instances where each dependency graph G_i is acyclic, and the in-degree of every issue in G_i is at most one. 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. Observe for example that an acyclic dependency graph for every voter does not necessarily lead to an acyclic global dependency graph. Furthermore, if each G_i is of bounded degree, this does not imply a constant upper bound for the maximum 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 certain interesting classes of graphs, as summarized in Theorem 3.

We first exhibit a property that allows us to reduce the solution of certain instances to the solution of instances with smaller sets of issues. Given a directed graph (V, E), the *neighborhood* of a vertex u is the set of its in-neighbors and out-neighbors: $N(u) = \{v \in V : (u, v) \in E \text{ or } (v, u) \in E\}$.

Lemma 3. *Consider a profile* P*, where for every voter* i*,* $\Delta_i \leq 1$*, and let* G *be the global dependency graph of* P*. If* G *has a vertex* y *with* $|N(y)| \leq 2$*, we can modify* P *in polynomial time to a profile* P' *(maintaining that every voter has maximum in-degree at most* 1*) with global dependency graph* H*, such that* $V(H) = V(G) \setminus \{y\}$*, and* CMS *on* P *is reduced to optimally solving* CMS *on* P'*.*

Proof. Fix a profile $P = (I, [n], B)$ with the aforementioned properties. For notational convenience in the proof, for every issue $x \in I$, we let $\{x_0, x_1\}$ be its domain, and recall that B_i^x denotes the projection of voter i's ballot on issue x.

We will first introduce a cost function that helps us decompose the total number of disagreements by an assignment of values to issues. Namely, for any directed edge (x, y) in the global dependency graph, and every assignment of values, say x_i, y_j, to these two issues, we let $c(x_i, y_j) = |\{v \in [n] : (x, y) \in E(G_v), \{x_i : y_j\} \notin B_v^y\}|$. In words $c(x_i, y_j)$ is the number of voters who have expressed a conditional vote for issue y, dependent only on x, and at the same time are dissatisfied with issue y, by the assignment (x_i, y_j). In addition, we set $c(y_j) = |\{v \in [n] : N_v^-(y) = \emptyset, y_j \notin B_v^y\}|$. Thus, $c(y_j)$ is the number of voters who have expressed an unconditional vote on y, and are dissatisfied with the value y_j.

Let us now consider the following three cases for issue y:

Case 1: If $|N(y)| = 0$, all votes for issue y are unconditional. Let P' be the profile that results after deleting vertex y. Then $\text{OPT}(P) = \text{OPT}(P') + \text{OPT}(y)$ where the optimal choice for y is the value that causes the least number of disagreements.

Case 2: If $|N(y)| = 1$, to get rid of vertex y, we keep track of the optimal choice for y under the possible values for its in-neighbor, say x. WLOG, we examine the case where both directed edges $(x, y), (y, x)$ appear in G. If one of these edges is not present, one just needs to adjust accordingly Equation (2) below. The fact that y does not have any dependencies with any issue other than x, allows us to compute its optimal value, given an assignment for x. Namely, for $i \in \{0, 1\}$, we compute and store the following quantity along with the corresponding value of y.

$$c^*(x_i) = \min_{k\in\{0,1\}} \{c(y_k) + c(x_i, y_k) + c(y_k, x_i)\}. \tag{2}$$

In case the minimum is achieved in (2) by both values of y, we can select one of them, according to some consistent tie-breaking rule. Hence, at the moment, we know how to set y, if we are given the value of x. Also, it is important to note that even if vertex x has in-degree higher than 2 in G, we have assumed that the maximum in-degree in every voter's dependency graph is at most one, and no voter would need to look at the value of y in combination with other issues to decide if she is satisfied with x. Thus we can leave y aside without causing any problems.

To proceed we produce a new profile P', from P as follows: *(i)* We delete issue y from I and from the dependency graphs. For every voter we also delete her expressed preferences for y, whether conditional or not. *(ii)* For every $i \in \{0, 1\}$, we introduce $c^*(x_i)$ new voters who are dissatisfied only with the assignment x_i of x, and are satisfied with any assignment on other issues.

It is easy to see that the global dependency graph of the newly created profile P' is exactly G without y and its adjacent edges. To complete the proof we have to argue that the value of the optimal solution in P' is the same as in the original instance. We defer this argument to Claim 2.

Case 3: If $|N(y)| = 2$, suppose that issue y is connected to issues x and z. As in *Case 2*, WLOG, assume that all edges $(x, y), (y, x), (y, z), (z, y)$ appear in the global dependency graph. The fact that there are no dependencies between y and any issues other than x and z, allows us to compute the optimal alternative for y, given an assignment of values to issues x and z. In analogy to Equation (2), for every $i, j \in \{0, 1\}$, we compute and store a quantity which expresses the minimum number of disagreements that can be caused by issue y, when we fix x to x_i and z to z_j. Namely, $c^*(x_i, z_j)$ equals

$$\min_{k\in\{0,1\}} \{c(y_k) + c(x_i, y_k) + c(y_k, x_i) + c(z_j, y_k) + c(y_k, z_j)\}.$$

We now produce a new profile P' from P as follows:

- We delete y from I and from the dependency graphs. We delete also each voter's expressed preferences for y.

- For every voter who had a conditional ballot on x, dependent on y, we replace it with the unconditional ballot $\{x_0, x_1\}$, i.e., the voter is now satisfied with any outcome on x. We do the analogous replacement for voters who had a conditional ballot on z dependent on y.
- For every $i, j \in \{0, 1\}$, we introduce $c^*(x_i, z_j)$ new voters, who have a conditional ballot for issue z, dependent on x. They are dissatisfied only with (x_i, z_j) and satisfied with any assignment on other issues.

It turns out that the global dependency graph of P', say H, is obtained from G by deleting y and its adjacent edges, and by adding the edge (x, z), if it was not already present. The proof can now be completed by the following claim.

Claim 2. *For the constructions of Cases 2 and 3, every solution of P corresponds to a solution of P' with the same cost and vice versa. Hence OPT$(P) =$ OPT(P').*

To argue about complexity, observe that we add at most $\mathcal{O}(n)$ new voters in moving from P to P'. Also when we solve P', to get an assignment for issue y of P, we need to remember either the values $\arg\min c^*(x_i)$ from (2) or in *Case 3*, the values $\arg\min c^*(x_i, z_j)$. □

Remark. *One can generalize the construction of Lemma 3 for vertices y with $|N(y)| = 3$. But the resulting profile P' may end up with voters of maximum in-degree two in their dependency graph. This prohibits a repeated use of Lemma 3, that we need in the sequel for obtaining optimal algorithms.*

We can now obtain positive results for concrete classes of graphs. We first introduce some more graph-theoretic terminology. Given a directed graph G, we refer to its *undirected version* as the undirected graph $\overline{G}$ produced after we remove the orientation of every edge of G. Furthermore, if for a pair of vertices x, y, both (x, y) and (y, x) are present in G, then we just keep a single edge (x, y) in $\overline{G}$.

In the next theorem, we identify a class of instances that admit an optimal solution in polynomial time, based on the undirected version $\overline{G}$ of the global dependency graph G of a given profile. Namely, the class consists of instances where $\overline{G}$ has treewidth at most 2. The treewidth is a parameter identifying how *close* to a tree a graph looks like. For the exact definition, we refer to [Robertson and Seymour, 1986]. The class of instances captured by our result includes paths, trees, cycles, series-parallel graphs, or any collection of such connected components. Further interesting classes that are included are cactus graphs, and ladder graphs.

Theorem 3. *If the dependency graph of every voter has maximum in-degree at most 1 and the undirected version of the global dependency graph has treewidth at most 2, then* CMS *is optimally solvable in polynomial time*[4].

Proof. Let $\overline{G}$ be the undirected version of the global dependency graph of a given profile. WLOG we may assume that $\overline{G}$ is connected, since otherwise we could solve for each connected component separately.

For the case where $\overline{G}$ has treewidth at most 1, then $\overline{G}$ is a tree and hence, it is possible to apply Lemma 3, so as to delete leaves sequentially until the remaining graph consists only of a single vertex. An optimal pick for that vertex is now possible and by backtracking, we can deduce the outcome for every issue in the optimal solution.

If $\overline{G}$ has treewidth equal to 2, there exists at least one vertex y with $|N(y)| \leq 2$ [Bodlaender, 1998], and hence, we can apply Lemma 3 again. The operations performed, when applying Lemma 3, to reduce the problem to a smaller instance cannot increase its treewidth (they involve deletions and contractions of edges and vertices). Thus, by successively applying the described procedure on the remaining instance, we end up with a graph of constant size (or even a single vertex), where CMS can be computed efficiently. □

We argue that some of the graph classes captured by Theorem 3 are meaningful in multi-issue elections with logically dependent issues. First, consider the case where the global dependency graph is a path with all edges oriented in the same way. We can think of the issues as ordered on a line, which is very natural when there is sequential dependence along a series of decisions. For an example, a municipality may need to decide on 3 issues: what public project to implement (say a park or a stadium), in which location to do it (dependent on the type of project, since each location can have different features), and how to connect it with existing means of public transportation (via a new bus stop or metro stop, dependent on location). Similarly, when the global dependency graph forms a directed tree oriented from the root towards the leaves, we again have a hierarchy regarding dependencies. E.g., a star graph oriented towards the leaves can arise when the senior partners of a firm have to decide on the location for a new subsidiary. This choice prominently affects a set of other decisions like the suppliers, marketing strategies, etc.

5 Conclusions and Future Work

We advocate that CMS combines a higher level of expressiveness with efficient algorithms for several cases of interest. We find the assumption of bounded in-degree as a motivated starting point for studying the computational properties of CMS, with several open questions remaining unresolved. Obtaining improved approximation ratios or inapproximability bounds for the cases we studied would provide further insights. It is a very interesting question whether efficient algorithms exist for any constant treewidth of the global dependency graph. Since we only provided a sufficient but not necessary condition for efficient algorithms, identifying parameters other than the treewidth, that would lead to optimal algorithms is also an intriguing topic. Finally, one can consider other objective functions, such as the Conditional Minimax rule, defined in [Barrot and Lang, 2016] or even non-binary domains. Algorithmic results there still remain elusive.

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".

[4]We are grateful to the anonymous IJCAI '20 reviewers for suggesting the condition on the treewidth, which yields a generalization of the results we claimed in the previous version of this work.

References

[Airiau *et al.*, 2011] 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)*, pages 18–23, 2011.

[Amanatidis *et al.*, 2015] 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.

[Avidor and Zwick, 2005] Adi Avidor and Uri Zwick. Approximating MIN 2-SAT and MIN 3-SAT. *Theory of Computing Systems*, 38(3):329–345, 2005.

[Barrot and Lang, 2016] 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.

[Bazgan and Paschos, 2003] Cristina Bazgan and Vangelis Th. Paschos. Differential approximation for optimal satisfiability and related problems. *European Journal of Operational Research*, 147(2):397–404, 2003.

[Bodlaender, 1998] Hans L Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theoretical Computer Science*, 209(1-2):1–45, 1998.

[Boutilier *et al.*, 2004] 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.

[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.

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

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

[Chevaleyre *et al.*, 2008] 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.

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

[Dalla Pozza *et al.*, 2011] 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)*, pages 172–177, 2011.

[Demange *et al.*, 1998] Marc Demange, Pascal Grisoni, and Vangelis Th. Paschos. Differential approximation algorithms for some combinatorial optimization problems. *Theoretical Computer Science*, 209(1-2):107–122, 1998.

[Gonzales *et al.*, 2008] 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.

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

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

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

[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.

[Lang *et al.*, 2018] Jérôme Lang, Jérôme Mengin, and Lirong Xia. Voting on multi-issue domains with conditionally lexicographic preferences. *Artificial Intelligence*, 265:18–44, 2018.

[Li *et al.*, 2010] 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*, pages 578–580, 2010.

[Marathe and Ravi, 1996] Madhav V. Marathe and S.S. Ravi. On approximation algorithms for the minimum satisfiability problem. *Information Processing Letters*, 58(1):23–29, 1996.

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

[Vazirani, 2003] Vijay V. Vazirani. *Approximation algorithms*. Springer Science & Business Media, 2003.

[Williamson and Shmoys, 2011] David P. Williamson and David B. Shmoys. *The Design of Approximation Algorithms*. Cambridge University Press, 2011.

[Xia and Conitzer, 2012] 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.