Conditional and Sequential Approval Voting on Combinatorial Domains

N. Barrot, J. Lang

Abstract

Several methods exist for making collective decisions on a set of variables when voters possibly have preferential dependencies. None is based on approval voting. We define a family of rules for approval-based voting on combinatorial domains, where voters cast *conditional approval ballots*, allowing them to approve values of a variable conditionally on the values of other variables. We study three such rules. The first two generalize simple multiwinner approval voting and *minimax* approval voting. The third one is an approval-based version of sequential voting on combinatorial domains. We study some properties of these rules, and compare their outcomes.

1 Introduction

Collective decisions on combinatorial domains cover a large variety of common situations, such as finding a set of dates for a series of meetings, electing a committee of representatives, adopting or rejecting each of a series of yes-no questions about common facilities to be built, or finding a common set of movies for a group to watch together.

Several methods have been proposed and studied for making such collective decisions. An important source of problems is when voters have *preferential dependencies* between variables: someone may be willing to attend the second meeting on a Friday provided that the first meeting is not already on a Friday; someone may want Ann to be elected at the department council only if both Betty and Charles are not elected too (they all belong to the same research group and would have too much joint power); I may want to watch one comedy and one drama, but not two films of the same genre.

It has been argued many times that making decisions independently on each variable cannot take such preferential dependencies into account. If we want to deal with them, we have to allow voters to express these dependencies. Several classes of methods have been proposed and studied (see [24] for a recent survey). *Direct* methods work by asking voters a one-shot report of their preferences, expressed in some language that allows preferential dependencies to be (at least partly) specified; *sequential* methods proceed by eliciting the voters' preferences on a first (fixed) variable, decide its value, broadcast it, then eliciting the voters' preferences on a second variable conditionally on the value chosen for the first one, and so on.

Most of the existing methods for voting in combinatorial domains assume that voters express, implicitly or explicitly, *rankings* over the values of variables, or else *cardinal utility functions*. Both may induce a cognitive burden to the voters, as well as an increased complexity in communication and computation. On the other hand, in simple domains, *approval voting* is a well-studied method for elections, where each voter may approve of any number of candidates, and the candidate with the largest number of approvals wins. Approval voting has nice properties, and is easy to use and understand [9].

Approval voting can be extended to combinatorial domains, and especially to committee elections. The simplest way, if we want to select k candidates, consists in selecting those with the largest k approvals. Other ways of extending approval voting to multiple winner elections have been considered and are surveyed in [22]. Similar methods can be used for multiple referenda, as argued in [2], due to the structural similarity of committee elections and multiple referenda: in both cases, a decision has to be made over a set of possible subsets (of candidates in one case, of binary issues in the other case) subject to feasibility constraints (such as the number of winners in committee elections, or budget constraints in multiple referenda). However, the methods we know of for multi-winner approval voting do not allow voters to express conditional preferences; they must vote as if they had separable preferences.

We propose a family of methods for reconciliating the simplicity of approval voting with the possibility for voters to express preferential dependencies. In Section 3 we define *conditional approval ballots* where voters approve some values of variables conditionally on the value of some other variables. In Section 4 we define two first rules based on conditional approval ballots, where the winning alternative minimizes respectively the maximum or the sum, over all voters, of the number of 'disagreements', and we show that these rules generalize simple (or 'minisum') approval voting and *minimax* approval voting [12]. In Section 5 we define *sequential conditional approval voting*: given a fixed order of variables, the voters approve some of their values, one variable after the other. We conclude in Section 6.

2 Background and related work

In a *committee* (or *multi-winner*) election, there is a set of candidates C, a set of voters N, each casting a vote (or a *ballot*), and a number k of winners (or sometimes a more flexible constraint, or even no constraint at all such as in Hall-of-Fame elections). In multiwinner approval ballotting, each voter casts an *approval ballot* where she approves as many candidates as she wants (in some variants, there are also constraints on the number of approvals on a ballot).

In *simple multiwinner approval voting*, the winners are simply the k candidates approved most often. Simple approval voting can be criticized for failing to guarantee a sufficient level of fairness or representativeness, and a number of other rules that map a collection of approval ballots to a set of winners have been studied; see [22] for a survey. One prominent such rule is *minimax approval voting*, where the elected committee minimizes the maximum, over all voters, of the number of 'disagreements' (*i.e.*, formally, the Hamming distance) between the voter's ballot and the chosen committee [12]. Unlike simple approval voting, minimax approval voting is NP-hard [30] and manipulable, but it can be efficiently approximated [26, 16, 15]; see [29] for its parameterized complexity. [2] generalize these two rules to other aggregation functions, and [5] extend them beyond approval ballots.

Other multiwinner approval voting rules, such as *proportional* or *satisfaction approval voting* [11], have been recently studied from the point of view of computation [4] or of their properties [3]. A number of works address the computation of *full proportional representation* with approval ballots, two good representatives being [31] and [33]. The computational aspects of strategic behaviour for single- and multi-winner approval voting are adressed in [28] and in [6]. The properties of multiwinner voting rules are studied more generally in [20].

Combinatorial domains are sets of alternatives consisting of the Cartesian product (or sometimes, a subset of it) of finite domain values corresponding to issues, variables, attributes, seats, or even individuals (in the case of committee elections). The main difficulty in *voting in combinatorial domains* is the presence of preferential dependencies. To deal with them, some approaches make use of compact representations, such as [32, 17, 27]. Some others make use of sequential voting, variable after variable, such as [23, 19, 1]. None of them makes use of approval ballots. See [24] for a recent overview of voting in combinatorial domains.

3 Conditional approval ballots

Let $\mathcal{X} = \{X_1, \ldots, X_p\}$ be a set of *variables*, each of them associated with a finite *value domain* D_i. Let $D = D_1 \times \ldots \times D_p$, and $D^* \subseteq D$ be a set of *feasible alternatives* (by default, $D^* = D$). A group of voters have to decide of a common outcome in D^*. When X_i is binary, we will usually take $D_i = \{x_i, \overline{x_i}\}$. For $J \subseteq \{1, \ldots, p\}$ we note $D_J = \times_{j \in J} D_j$, and for $\mathbf{d} = (d_1, \ldots, d_p) \in D$, d_J denotes the tuple $\mathbf{d}$ projected to the indices in J.

For instance, in a *designated-post committee election* [7], $\mathcal{X}$ is a set of different seats to be filled and their domains are the respective sets of candidates for each position (an alternative is feasible only if it does not assign the same candidate to more than one seat). In a standard committee election where the size of the committee is fixed to k, $\mathcal{X}$ can be identified with the set of candidates, with binary domains {elected, not elected}, and an alternative is feasible iff k variables have value 'elected'. In a multiple referendum [10], $\mathcal{X}$ is a set of binary issues.

As a running example, a group of friends have to decide on a common menu. The variables are X_1 (main dish), with $D_1 = \{m, f, v\}$ (meat, fish, vegetarian dish), and X_2 (drink), with $D_2 = \{r, w, b\}$ (red wine, white wine, beer).

The most common way of using approval ballotting in such domains consists in asking voters to approve values for each variable separately. This does not require much communication, but it is not possible for voters to report conditional preferences. For instance, a voter may approve $\{m, f\} \subseteq D_1$ and $\{r, w\} \subseteq D_2$ but cannot express that he approves the red wine only if the main dish collectively chosen is meat.

One could also think of asking voters to specify all combinations of values they approve (possibly in a compact way by means of a propositional formula). Here, preferential dependencies can be expressed, but the communication is costly, and moreover, a problem is that once a voter obtains a value he does not like for a variable, his opinion about the other variables does not count. Assume for instance that a voter hates fish and beer, and wants red wine with meat. It is reasonable to expect that he approves the set of menus $\{mr, vw, vr\}$, and disapproves mw, mb, fr, fw, fb, vb. However, if the collective decision turns out to give the value fish to X_1, our voter won't have a chance to express that he still does not want to drink beer.

A trade-off between these two (extreme) methods consists in allowing voters to *approve sets of values for each variable, conditioned on the value of some other variables.*

Definition 1. *A* conditional approval (CA) ballot *over variables $X_1, \ldots, X_p$ with domains $D_1, \ldots, D_p$ is a pair*

$$B = \langle G, \{A_i \mid i = 1, \ldots, p\} \rangle$$

where G is a directed graph over $\{X_1, \ldots, X_p\}$, and for each i, A_i is a set of conditional approval statements $\{u : a_i(u) | u \in D_{Par_G(X_i)}\}$*, where $a_i(u) \subseteq D_i$, and $Par_G(X_i)$ is the set of parents of X_i in G. $B^{\downarrow X_i | u} = a_i(u)$ denotes the* projection *of B over X_i conditioned by u.*

We say that B is acyclic *if G is acyclic. B is a* separable *CA ballot if G has no edges.*

Most often, we will alleviate the notation and only write the CA statements (and not the graph) when we specify a CA ballot. Also, we omit some curly brackets when it cannot lead to any confusion (in particular, for singletons).

Definition 2. *Given an CA ballot B and an assignment $\mathbf{d} = (d_1, \ldots, d_p) \in D_1 \times \ldots \times D_p$, we say that $\mathbf{d}$ disagrees with B on variable X_i if $d_i \notin B^{\downarrow X_i | d_U}$, where $U = Par_G(X_i)$. The disagreement of $\mathbf{d}$ with respect to B, denoted by $\delta(\mathbf{d}, B)$, is the number of variables X_i on which $\mathbf{d}$ disagrees with B.*

Example 1. *Consider a voter expressing the CA ballot*

$$B = \langle [X_1 \to X_2],\ \ \{\{f, v\};\ \ m : r;\ \ f : w;\ \ v : \{r, w\}\} \rangle$$

This voter has a preference for not having meat: $B^{\downarrow X_1} = \{f, v\}$; but in case the collective decision is to have meat, she still has her word to say about the drink, and approves (only) the red wine: $B^{\downarrow X_2 \mid m} = \{r\}$. If the collective decision is the vegetarian dish, then she approves both the red and the white wine: $B^{\downarrow X_2 \mid v} = \{r, w\}$. Now, $\mathbf{d} = mr$ disagrees with B on X_1 but not X_2, therefore $\delta(\mathbf{d}, B) = 1$ (she is not satisfied with the choice of the dish, but is satisfied by the choice of the *drink given that the dish is meat). Likewise, we have $\delta(fw, B) = \delta(vr, B) = \delta(vw, B) = 0$ (she is satisfied by the choice of the dish* and *the choice of the drink conditional on the value of the dish), $\delta(fr, B) = \delta(fb, B) = \delta(vb, B) = 1$, and $\delta(mw, B) = \delta(mb, B) = 2$.*

Equivalently, $p - \delta(\cdot, B)$ can be seen as a utility function which counts the number of conditional preferences satisfied. Thus, even though voters are using approval-like ballots, the preferences they express *are not dichotomous*.

Observation 1. *If B is acyclic and $a_i(u) \neq \emptyset$ for all $i \leq p$ and $u \in D_{Par_G(X_i)}$, then $\delta(\mathbf{d}, B) = 0$ for some $\mathbf{d} \in D$.*

This, however, is no longer true if B is not acyclic: let for instance $B = \{x_1 : \overline{x_2},\ \overline{x_1} : x_2,\ x_2 : x_1,\ \overline{x_2} : \overline{x_1}\}$, then for every $\mathbf{d} \in D$ we have $\delta(\mathbf{d}, B) = 1$.

Conditional approval ballots have a clear resemblance with CP-nets [8]. For the sake of simplicity, assume that variables are binary and let $N(B)$ the CP-net obtained by replacing all conditional approval sets $u : a_i(u)$ by $u : a_i(u) \succ \overline{a_i(u)}$, with $\overline{\overline{x_i}} = x_i$. The semantics of B and $N(B)$ are different: if $\succ_{N(B)}$ is the partial order over D induced from $N(B)$, and $\succ_B$ the order defined by $\mathbf{d} \succ_B \mathbf{d}'$ if $\delta(\mathbf{d}, B) < \delta(\mathbf{d}', B)$, then $\succ_B$ and $\succ_{N(B)}$ are incomparable:

Observation 2. *There exists B and $\mathbf{d}, \mathbf{d}'$ such that $\mathbf{d} \succ_B \mathbf{d}'$ and $\mathbf{d}' \succ_{N(B)} \mathbf{d}$.*

This can be seen on this example: $\mathcal{X} = \{X_1, X_2, X_3\}$; $B = \{x_1, x_1 : x_2, \overline{x_1} : \overline{x_2}, x_1x_2 : x_3, x_1\overline{x_2} : \overline{x_3}, \overline{x_1}x_2 : \overline{x_3}, \overline{x_1x_2} : x_3\}$. We have $x_1\overline{x_2}x_3 \succ_{N(B)} \overline{x_1x_2}x_3$, because $N(B)$ contains $x_1 \succ \overline{x_1}$, however $x_1\overline{x_2}x_3 \prec_B \overline{x_1x_2}x_3$, because $\delta(\overline{x_1x_2}x_3, B) = 1 < \delta(x_1\overline{x_2}x_3, B) = 2$.

Note that $\delta(\cdot, B)$ is a function from D to $[0, p]$, but not all functions from D to $[0, p]$ can be expressed by a CA ballot; this can be seen by a simple counting argument. Therefore, assuming that a voter's preferences are representable by a CA ballot is a domain restriction.

4 Conditional minisum and minimax

Definition 3. *An n-voter* conditional approval profile *is a collection $P = \langle B_1, \ldots, B_n \rangle$ of CA ballots. It is* separable *if every B_i is a separable CA ballot. A CA irresolute rule is a function mapping any CA profile to a nonempty subset of D^*.*

Resolute rules can be obtained by combining an irresolute rule with a tie-breaking mechanism: if F is an irresolute rule and T is a tie-breaking mechanism induced by a fixed priority ranking over D^*, then F^T is the composition of F by T. (Note that T is a relation over an exponentially large set, and thus should be represented in some compact way; this has no impact in our results.) We now define two specific rules.

4.1 Conditional minisum

Definition 4. *Given a CA profile $P = \langle B_1, \ldots, B_n \rangle$, the* conditional minisum *rule outputs the outcomes that minimize the total number of disagreements over all voters:*

$$CondMiniSum(P) = \text{argmin}_{\mathbf{d} \in D^*} \sum_{i=1}^{n} \delta(\mathbf{d}, B_i)$$

Example 2. *Let $D^* = D = \{m, f, v\} \times \{r, w, b\}$ and let P be the following 19-voter CA profile; the number on the top row denotes the number of voters; for instance, 5 voters express the CA ballot*

$\{\{m, f\}, m : r, f : w, v : \{r, w\}\}$. *Note that the CA ballots of the second column are separable.*

5	4	4	3	3
$\{m, f\}$	m	$\{m, f, v\}$	$\{f, v\}$	v
$m : r$	$m : b$	$m : \{r, w\}$	$m : r$	$m : b$
$f : w$	$f : b$	$f : w$	$f : w$	$f : b$
$v : \{r, w\}$	$v : b$	$v : \{r, b\}$	$v : \{r, w\}$	$v : \{r, w, b\}$

The table below gives the disagreement values for each group of voters, and the total disagreement value.

	mr	mw	mb	fr	fw	fb	vr	vw	vb
5	0	1	1	1	0	1	1	1	2
4	1	1	0	2	2	1	2	2	1
4	0	0	1	1	0	1	0	1	0
3	1	2	2	1	0	1	0	0	1
3	2	2	1	2	2	1	0	0	0
total	13	21	18	26	14	19	13	17	17

and we have $CondMiniSum(P) = \{mr, vr\}$.

If P is separable, then the outcome of $CondMiniSum(P)$ can be obtained simply by decomposing the vote into a series of standard approval votes, variable by variable. Let C be a set of candidates; a (standard) approval profile is a collection of approval ballots $(A_1, \ldots, A_n)$ where $A_i \subseteq C$ for each i; the (standard) approval rule *App* maps any approval profile to a winner (or a set of tied winners), and is defined by $App(A_1, \ldots, A_n) = \text{argmax}_{c \in C} |\{i : c \in A_i\}|$.

Observation 3. *If* $P = \langle B_1, \ldots, B_n \rangle$ *be a separable CA profile. Then* $CondMiniSum(P) = App(P^{\downarrow X_1}) \times \ldots \times App(P^{\downarrow X_p})$*, where* $P^{\downarrow X_i} = (B_1^{\downarrow X_i}, \ldots, B_n^{\downarrow X_i})$.

In particular, if all variables are binary, we are in the context of a multiple referendum without preferential dependencies between issues, and where the decision is made issue by issue according to majority.

On the other hand, classical committee elections correspond to a collection of such separable conditional approval ballots, where binary variables correspond to candidates, together with a constraint on the cardinality of the committee.

Observation 4. *Let* $D = \{yes, no\}^p$ *(plus possibly a cardinality constraint), and* $P = \langle B_1, \ldots, B_n \rangle$ *a separable CA profile. Then* $CondMiniSum(P)$ *coincides with the output of simple multiwinner approval voting (also called* minisum *in [12]).*

4.2 Conditional minimax

Conditional minisum has a utilitarianistic flavor: minimizing the sum of disagreements corresponds to maximizing social welfare. We may instead take an egalitarian point of view and minimize the *maximum* disagreement:

Definition 5. *Given a CA profile* $P = \langle B_1, \ldots, B_n \rangle$*, the* conditional minimax *rule outputs the outcomes that minimize the maximum number of disagreements over all voters :*

$$CondMiniMax(P) = \text{argmin}_{\mathbf{d} \in D^*} \max_{i=1}^{n} \delta(\mathbf{d}, B_i)$$

Example 2, continued. *The only outcome that has a disagreement at most 1 (in fact, exactly 1) with every conditional ballot is* fb*, therefore* $CondMiniMax(P) = \{fb\}$*: every agent is either happy with the fish, or unhappy with the fish, but in that case, happy with the beer given that the dish is fish.*

A similar result as Observation 4 holds for conditional minimax:

Observation 5. *Let $D = \{yes, no\}^p$ (plus possibly a cardinality constraint), and $P = \langle B_1, \ldots, B_n \rangle$ a separable CA profile. Then $CondMiniMax(P)$ is the output of* minimax *approval voting [12]).*

4.3 Computation

Computing a winning committee for minimax approval voting is NP-hard [30], which of course carries over to conditional minimax approval voting. On the other hand, the polynomial time computation of winning committees for simple approval voting does not carry over to conditional minisum (even without feasibility constraints):

Proposition 1. *Given a CA profile P and an integer k, deciding whether there exists $\mathbf{d} \in D$ such that $\sum_i \delta(\mathbf{d}, B_i) \leq k$ is* NP*-complete, even for binary variables and, for each voter, an acyclic dependency graph with maximal indegree 1.*

Proof. The problem is clearly in NP. The hardness proof is based on a reduction from MAX2SAT. Consider an instance I of MAX2SAT, defined by a set of variables $V = \{x_1, \ldots, x_p\}$, a set of 2-clauses $C = \{C_1, \ldots, C_n\}$, and an integer k. We create a set of binary variables $\mathcal{X} = \{X_1, \ldots, X_p\}$ and, for each clause $C_i = l_j \vee l_k$, where $l_j \in \{x_j, \overline{x_j}\}$ and $l_k \in \{x_k, \overline{x_k}\}$, and $j < k$, a voter i whose CA ballot B_i is defined as follows (with the convention $\overline{l_j} = x_j$ when $l_j = \overline{x_j}$):

- the dependency graph has a single edge $X_j \longrightarrow X_k$.
- the CA statements are $\{x_j, \overline{x_j}\}$, $l_j : \{x_k, \overline{x_k}\}$, $\overline{l_j} : \{l_k\}$, and $\{x_q, \overline{x_q}\}$ for each $q \neq k$.

First, assume that there exists an assignment of the variables V satisfying at least k clauses. From that assignment, we define an alternative $\mathbf{d}$ as follows: for all $q \leq p$, $\mathbf{d}_q = x_q$ if x_q is assigned to true, and $\mathbf{d}_q = \overline{x_q}$ if x_q is assigned to false. Then, we study the number of disagreements between B_i and $\mathbf{d}$. There are two cases to consider:

- If C_i is satisfied, then either l_j is true or l_k is true, and from the definition of B_i, we get $\delta(\mathbf{d}, B_i) = 0$.
- If C_i is not satisfied, then both l_j and l_k are false, and from the definition of B_i, we get $\delta(\mathbf{d}, B_i) = 1$.

Thus, since the assignment satisfies at least k clauses, we have $\sum_{i=1}^{n} \delta(\mathbf{d}, B_i) \leq p - k$.

Conversely, if there exists a $\mathbf{d}$ such that $\sum_{i=1}^{n} \delta(\mathbf{d}, B_i) \leq p - k$, then by a similar line of reasoning as above, we can construct an assignment that satisfies at least k clauses. □

On the positive side, there is a model-preserving translation from winner determination for conditional minisum approval voting into maximum satisfiability. We omit the formal details of the general construction but only give an example: the CA ballot $B = \{x,\ x : y,\ \overline{x} : \{y, \overline{y}\}\}$ is translated into the set of clauses $\mathcal{C}_B = \{x, \overline{x} \vee y, x \vee y \vee \overline{y}\}$ (equivalent to $\{x, \overline{x} \vee y\}$); then a CA profile $P = \langle B_1, \ldots, B_n \rangle$ is translated into a multi-set of clauses $\mathcal{C}_P = \mathcal{C}_{B_1} \cup \ldots \cup \mathcal{C}_{B_n}$. For all $\mathbf{d} \in D$, we have $\sum_i \delta(\mathbf{d}, B_i) = \alpha$ if and only if the number of clauses that are not satisfied in $\mathcal{C}_P$ is equal to α.[1]

This means that conditional minisum approval voting can be solved by off-the-shelf MAXSAT solvers. Moreover, in the case of binary variables with acyclic dependency graphs, it is easy to check that this transformation preserves the differential approximation ratio of $4.34/(m + 4.34)$ obtained for MAXSAT in [21], where m represents the number of clauses.[2]

[1] The translation works also for nonbinary variables, but is more complex, as there is one propositional variable per value $d_i \in D_i$, and a feasibility constraint ensures that each variable takes exactly one value. More generally, when D^* is a strict subset of D, the feasibility constraints are expressed as propositional formulas, and are in sufficiently many copies so as to be ‘protected’.

[2] Given an instance I of a combinatorial optimization problem, the differential approximation ratio measures “the relative

4.4 Properties

We now study some of the properties satisfied by conditional minisum and conditional minimax. Because our rules have CA ballots as input, some of the standard properties have to be adapted. Clearly, our rules satisfy anonymity (the outcome is independent from the identity of voters), and their irresolute versions satisfy value neutrality (the outcome is unchanged after any renaming of the values of some variable) and variable neutrality (the outcome is unchanged after any renaming of the variables). Now we define four other important properties. For two profiles $P = (P_1, \dots, P_n)$ and $Q = (Q_1, \dots, Q_m)$, we note $P + Q = (P_1, \dots, P_n, Q_1, \dots, Q_m)$.

Definition 6. *Let F be a CA irresolute rule.*

- *F satisfies* reinforcement *(resp.* weak reinforcement*) if for any two profiles P, Q, if $F(P) \cap F(Q) \neq \emptyset$ then $F(P + Q) = F(P) \cap F(Q)$ (resp. $\supseteq F(P) \cap F(Q)$).*
- *F satisfies* monotonicity *if for any profile $P = (B_1, \dots, B_n)$, if $\mathbf{d} = (x_1, \dots, x_p) \in F(P)$ and B'_i is obtained from B_i by adding x_j to the set of values of X_j approved conditioned on $d_{Par_{G_i}(X_j)}$, then $\mathbf{d} \in F(B_1, \dots, B_{i-1}, B'_i, B_{i+1}, \dots, B_n)$.*

The following properties, participation and strategyproofness, are defined only for resolute rules.[3] Moreover, they require a voters' preference $\succeq_i$ to be induced from her CA ballot. We say that a voter with CA ballot B_i has *δ-induced preferences* if for all $x, y \in D^*$, $x \succeq_i y$ if and only if $\delta(x, B) \leq \delta(y, B)$. δ-induced preferences generalize Hamming-induced preferences [18].

Definition 7. *Let F^T the resolute version of some irresolute rule F, for some tie-breaking mechanism T. F^T satisfies*

- *F^T satisfies* δ-participation *if for any CA profile $P = (B_1, \dots, B_n)$ and CA ballot B_{n+1}, we have $\delta(F^T(P + \{B_{n+1}\}), B_{n+1}) \leq \delta(F^T(P), B_{n+1})$.*
- *F^T satisfies* δ-strategyproofness *if for any CA profile $P = (B_1, \dots, B_n)$ and CA ballot B'_i, $\delta(F^T(B_1,\dots,B_{i-1},B'_i,B_{i+1},\dots,B_n), B_i) \geq \delta(F^T(P), B_i)$.*

We have the following results for conditional minisum and minimax. All proofs are simple (we omit them).

Proposition 2.

- *$CondMinSum$ satisfies reinforcement.*
- *$CondMinMax$ does not satisfy reinforcement, but satisfies weak reinforcement.*
- *$CondMinSum$ and $CondMinMax$ satisfy monotonicity.*
- *for any T, $CondMinSum^T$ and $CondMinMax^T$ satisfy δ-participation.*

Since minimax approval voting [16] is manipulable, this is a fortiori the case for $CondMinMax^T$. On the other hand, simple approval voting (or 'minisum') is strategyproof; however, as soon as preferences are conditional, strategyproofness is lost:

Proposition 3. *$CondMinSum^T$ is δ-manipulable, even for two binary variables.*

Proof. Without loss of generality, assume T favors $\overline{x_1 x_2}$ over $x_1 x_2$. Consider the two-voter CA profile consisting of one ballot $\{\overline{x_1},\ x_1 : x_2,\ \overline{x_1} : \overline{x_2}\}$ and one ballot $\{x_1, x_1 : x_2, \overline{x_1} : \overline{x_2}\}$. The outcome is $\overline{x_1 x_2}$. Now, if the second voter casts the ballot $\{x_1,\ x_1 : x_2,\ \overline{x_1} : x_2\}$ instead, then the outcome is $x_1 x_2$, which is her preferred alternative under the assumption that her preferences are δ-induced. □

position of the value of an approximated solution in the interval between the value of a worst feasible solution of I, and the value of a best solution of I" [21].

[3] For irresolute rules, we would require to first define a preference extension principle (see, *e.g.*, [14] for strategyproofness and [13] for participation).

5 Sequential conditional approval voting

For the sake of simplicity, in this section we assume that $D^* = D$ (handling feasibility constraints is possible but makes the definitions more complicated).

Sequential conditional approval voting (SCAV) is an approval-based version of sequential voting in combinatorial domains [23]. Its intuitive principle is that decisions are taken variable by variable, following a fixed order; for each variable, the winning value is the one that has the maximal approval score given the values of the variables that come before it. The SCAV rule is defined only for CA profiles whose dependency graph is compatible with a given order O of the variables. In the rest of this section, $O = X_1 > \ldots > X_p$ is a fixed order over $\mathcal{X}$.

Definition 8 (O-legal conditional approval profile)**.** *A CA ballot is O-legal if its dependency graph contains no edge from X_j to X_i with $i < j$. A CA profile is O-legal if it is composed of O-legal CA ballots.*

Definition 9. *The* sequential conditional approval voting rule *(SCAV) is the rule defined on O-legal CA profiles as follows. Let App^{T_i} be the standard approval voting rule on D_i, together with a tie-breaking mechanism T_i, and $T = (T_1, \ldots, T_p)$. For each O-legal $P = (B_1, \ldots, B_n)$, let $P^{\downarrow X_i | x_1^*, \ldots, x_{i-1}^*} = (B_1^{\downarrow X_i | x_1^*, \ldots, x_{i-1}^*}, \ldots, B_n^{\downarrow X_i | x_1^*, \ldots, x_{i-1}^*})$; then $SCAV^T(P) = (x_1^*, \ldots, x_p^*)$, where*

- $x_1^* = App^{T_1}(P^{\downarrow X_1})$;
- *for each* $i = 2, \ldots, p$, $x_i^* = App^{T_i}(P^{\downarrow X_i | x_1^*, \ldots, x_{i-1}^*})$.

Similarly, we define the SCAV *irresolute* rule: $SCAV(P)$ is the set of all $(x_1, \ldots, x_p)$ such that $x_1 \in App(P^{\downarrow X_1})$ and for all $i = 2, \ldots, p$, $x_i \in App(P^{\downarrow X_i | x_1^*, \ldots, x_{i-1}^*})$.[4][5]

A cheap protocol for $SCAV^T$ is composed of p rounds: at round i, it elicits only the voters' approval ballots $B_j^{\downarrow X_i | x_1^*, \ldots, x_{i-1}^*}$. Thus, the communication complexity of $SCAV^T$ is $O(pn \max_i |D_i|)$.

Example 2 (continued). *Let $O = X_1 > X_2$. The approval scores for m, f and v are respectively 13, 12 and 10: the (unique) selected value for X_1 is m. Given $X_1 = m$, the approval scores for r, w and b are respectively 12, 4 and 7: the selected value for X_2 is r. The outcome is mr.*

For irresolute SCAV, *some* winning alternative can be computed in polynomial time, and determining whether a given alternative is a winner is also polynomial-time computable: at each round, it suffices to check that $x_i \in App(P^{\downarrow X_i | x_1^*, \ldots, x_{i-1}^*})$. Also, winner determination is polynomial for resolute SCAV (with a polynomial-time computable tie-breaking mechanism).

Recall that winner determination for conditional minisum is NP-hard, even for an O-legal profile. Moreover, both conditional minisum and SCAV coincide with simple approval voting when restricted to separable profiles. Thus, we may wonder how good an approximation SCAV is to conditional minisum. The approximation ratio is different whether we measure the quality of an alternative by its number of disagreements $\sum_i \delta(\cdot, B_i)$ (to be minimized) or its number of agreements $\sum_i (p - \delta(\cdot, B_i))$ (to be maximized):

Proposition 4. *Consider a combinatorial domain with p variables $X_1, \ldots, X_p$ with $|D_i| = \alpha_i$ for all i.*

- *The largest possible ratio, over all O-legal CA profiles, between the disagreement score of the SCAV winner and the disagreement score of the conditional minisum winner, is* $\sum_{i=1}^{p}(1 - \frac{1}{\alpha_i})/(1 - \frac{1}{\alpha_1})$.

[4] Note that writing $B_j^{\downarrow X_i | x_1^*, \ldots, x_{i-1}^*}$ is a slight abuse of notation, since O-legality implies only that $Par_{G_j}(X_i) \subseteq \{X_1, \ldots, X_{i-1}\}$.

[5] Note that the restriction of SCAV to separable profiles coincides with simple approval voting.

- *The largest possible ratio, over all O-legal CA profiles, between the agreement score of the conditional minisum winner and the agreement score of the SCAV winner, is* $(\frac{1}{\alpha_1} + p - 1)/\sum_{i=1}^{p} \frac{1}{\alpha_i}$.

Proof sketch. Consider n CA ballots, with the dependency graph whose edges are $\{X_i \to X_j \mid 1 \leq i < j \leq p\}$. First, we look for an upper bound of the disagreement score of an alternative that wins for SCAV. Assume that $(x_1^1 x_1^2 \dots x_1^p)$ is the winning alternative for SCAV. Then, for $(x_1^1 x_1^2 \dots x_1^p)$ to be winning for SCAV, x_1^1, respectively $x_1^1 : x_1^2$, …, $x_1^1 x_1^2 \dots : x_1^p$, has to be approved by at least n/α_1 voters, respectively $n/\alpha_2, \dots, n/\alpha_p$ voters. It leads to a maximal disagreement score of $n \cdot \sum_{i=1}^{p} 1 - 1/\alpha_i$.

Now, we look for a lower bound for the disagreement score of a winning alternative for conditional minisum, under the condition that it is winning for SCAV. Assume that $(x_2^1 x_2^2 \dots x_2^p)$ is a winning alternative for conditional minisum. One way to avoid this alternative to be a SCAV winner is by eliminating it at the first round of the sequential vote. It implies that x_2^1 has to be approved by less than $\frac{n}{\alpha_1}$ voters, and, in this case, $x_2^1 : x_2^2, \dots, x_2^1 x_2^2 \dots : x_2^p$ can be approved by any number of voters, and then, the disagreement score for $(x_2^1 x_2^2 \dots x_2^p)$ without being a winner for SCAV is at least $n - \frac{n}{\alpha_1}$. There exist other ways to avoid $(x_2^1 x_2^2 \dots x_2^p)$ to be a winning alternative for SCAV, but they lead to disagreement scores that are at least as large as that one. The proof for the bound relative to the agreement score is similar.

Thus, we get a worst-case ratio of $(\sum_{i=1}^{p} 1 - 1/\alpha_i)/(1 - 1/\alpha_1)$. This ratio is reached on a CA profile for which there is a perfect split between all values of X_1 (with each voter approving only one value), and then, conditionally on the value chosen for X_1 (by tie-breaking), there is also a perfect split for each other variable, while for some of the other values of X_1, there is a perfect agreement between all voters on all other variables. The same profile shows that the bound relative to the number of agreements is reached too. □

In particular, when all domains have the same cardinality α, then these ratios become p and $(1 + (p - 1)\alpha)/p$, and in the case of binary domains, p and $2 - 1/p$.

Corollary 1. *If the quality of a solution is measured by the number of disagreements, then SCAV is a* $\sum_{i=1}^{p}(1 - \frac{1}{\alpha_i})/(1 - \frac{1}{\alpha_1})$*-approximation of* $CondMiniSum$*. If it is measured by the number of disagreements, then it is a* $(\frac{1}{\alpha_1} + p - 1)/\sum_{i=1}^{p} \frac{1}{\alpha_i}$*-approximation of* $CondMiniSum$*.*

Proposition 5. *SCAV satisfies anonymity, neutrality, reinforcement and monotonicity. For any tie-breaking mechanism T, SCAV^T does not satisfy δ-participation.*

Proof. Anonymity and neutrality are obvious. Reinforcement and monotonicity are easily proven by induction on the variables, and using the fact that standard approval voting satisfies them. For participation, consider the two-voter CA profile $P = \langle B_1, B_2 \rangle$ with $B_1 = \{\overline{x_1}, \overline{x_2}, \overline{x_3}\}$ and $B_2 = \{x_1, \overline{x_2}, \overline{x_3}\}$. Assume without loss of generality that T favors $\overline{x_1 x_2 x_3}$ over $x_1 \overline{x_2 x_3}$, thus $F^T(P) = \overline{x_1 x_2 x_3}$. Now, consider a third voter with preferences δ-induced by $B_3 = \{x_1, x_1 : x_2, \overline{x_1} : \overline{x_2}, x_1 : x_3, \overline{x_1} : \overline{x_3}\}$. If she votes, the outcome is $x_1 \overline{x_2 x_3}$, which has only one agreement with her ballot, while $\overline{x_1 x_2 x_3}$ had two. □

Proposition 6. *For any tie-breaking mechanism T, SCAV^T is not δ-strategyproof for $p \geq 3$ (even for binary variables), and is strategyproof for two variables.*

Proof. For $p = 3$, consider the profile in the proof of Proposition 5. If the third voter expresses his sincere ballot $\{x_1, x_1 : x_2, \overline{x_1} : \overline{x_2}, x_1 : x_3, \overline{x_1} : \overline{x_3}\}$, the outcome is $x_1 \overline{x_2 x_3}$. If he expresses instead the ballot $\{\overline{x_1}, x_1 : x_2, \overline{x_1} : \overline{x_2}, x_1 : x_3, \overline{x_1} : \overline{x_3}\}$ the the outcome is $\overline{x_1 x_2 x_3}$, which he prefers.

For $p = 2$, let $\mathcal{X} = \{X_1, X_2\}$; suppose that voter i with sincere ballot B has a manipulation B'; let x^* be the outcome for her sincere vote and x^+ the outcome after manipulation. Since B' is a manipulation and i has δ-induced preferences, we have $\delta(x^+, B') < \delta(x^*, B)$. This implies

(1) $\delta(x^+, B') < 2$ and (2) $\delta(x^*, B) > 0$. Once the value of X_1 has been fixed, we are left with a standard approval vote on X_2 and i cannot obtain a better outcome than the sincere outcome; therefore, the manipulation has to bear on X_1 and has to change the value of X_1, from, without loss of generality, x_1 to x'_1, with i initially approving x_1 but not x'_1. But this implies that $\delta(x^+, B') \geq 1$ and $\delta(x^*, B) \leq 1$, which together with (1-2) implies $\delta(x^+, B') = \delta(x^*, B) = 1$, therefore i, who has δ-induced preferences, cannot prefer x^+ to x^*. □

6 Conclusion

We have generalized approval voting to combinatorial domains and to nonseparable preferences. Our rules are natural generalizations of simple (or 'minisum') and minimax approval voting. They can be applied for all types of voting on combinatorial domains. Conditional minisum and minimax are computationally hard, but conditional minisum can be solved by MaxSAT solvers. Sequential conditional AV is easy to compute. Our rules (especially conditional minisum) satisfy a number of important properties.

There are other rules for multiwinner elections using approval ballots, where the satisfaction of a voter is not simply the Hamming distance to his preferred outcome (see [22] for a survey, and [4] for their computation). These rules could be adapted to CA ballots in the same way as we did for simple and minimax approval voting. The rules defined in [2], where the dissatisfaction degrees of the voters is aggregated by ordered weighted averages intermediate between max and sum, could also be generalized to CA ballots.

Acknowledgement

We thank Bernard Ries for helpful discussions. This research has been supported by the project ANR-14-CE24-0007-01 CoCoRICo-CoDec.

References

[1] S. Airiau, U. Endriss, U. Grandi, D. Porello, and J. Uckelman. Aggregating dependency graphs into voting agendas in multi-issue elections. In *IJCAI*, pages 18–23, 2011.

[2] G. Amanatidis, N. Barrot, J. Lang, E. Markakis, and B. Ries. Multiple referenda and multiwinner elections using Hamming distances: Complexity and manipulability. In *AAMAS*, pages 715–723, 2015.

[3] H. Aziz, M. Brill, V. Conitzer, E. Elkind, R. Freeman, and T. Walsh. Justified representation in approval-based committee voting. In *AAAI*, 2015.

[4] H. Aziz, S. Gaspers, J. Gudmundsson, S. Mackenzie, N. Mattei, and T. Walsh. Computational aspects of multi-winner approval voting. In *AAMAS*, pages 107–115, 2015.

[5] D. Baumeister, S. Dennisen, and L. Rey. Winner determination and manipulation in minisum and minimax committee elections. In *ADT*, pages 469–485, 2015.

[6] D. Baumeister, G. Erdélyi, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Computational aspects of approval voting. In *Handbook on Approval Voting*, chapter 10, pages 199–251. 2010.

[7] J.-P. Benoît and L. Kornhauser. Only a dictatorship is efficient. *Games and Economic Behavior*, 70(2):261–270, 2010.

[8] C. Boutilier, R. I. Brafman, C. Domshlak, H. H. Hoos, and D. Poole. CP-nets: A tool for representing and reasoning with conditional ceteris paribus preference statements. *J. Artif. Intell. Res. (JAIR)*, 21:135–191, 2004.

[9] S. Brams and P. Fishburn. Going from theory to practice: the mixed success of approval voting. *Social Choice and Welfare*, 25(2-3):457–474, 2005.

[10] S. Brams, D. Kilgour, and W. Zwicker. The paradox of multiple elections. *Social Choice and Welfare*, 15(2):211–236, 1998.

[11] S. Brams and M. Kilgour. Satisfaction approval voting. In *Voting Power and Procedures*, pages 323–346. Springer, 2014.

[12] S. Brams, M. Kilgour, and R. Sanver. A minimax procedure for electing committees. *Public Choice*, 132(3-4):401–420, 2007.

[13] F. Brandl, F. Brandt, C. Geist, and J. Hofbauer. Strategic abstention based on preference extensions: Positive results and computer-generated impossibilities. In *IJCAI*, pages 18–24, 2015.

[14] F. Brandt and M. Brill. Necessary and sufficient conditions for the strategyproofness of irresolute social choice functions. In *TARK*, pages 136–142, 2011.

[15] J. Byrka and K. Sornat. Ptas for minimax approval voting. In *Web and Internet Economics*, pages 203–217. Springer, 2014.

[16] I. Caragiannis, D. Kalaitzis, and E. Markakis. Approximation algorithms and mechanism design for minimax approval voting. In *AAAI*, 2010.

[17] V. Conitzer, J. Lang, and L. Xia. Hypercubewise preference aggregation in multi-issue domains. In *IJCAI*, pages 158–163, 2011.

[18] T. Çuhadaroğlu and J. Lainé. Pareto efficiency in multiple referendum. *Theory and decision*, 72(4):525–536, 2012.

[19] G. dalla Pozza, M. Pini, F. Rossi, and K. Venable. Multi-agent soft constraint aggregation via sequential voting. In *IJCAI*, pages 172–177, 2011.

[20] E. Elkind, P. Faliszewski, P. Skowron, and A. Slinko. Properties of multiwinner voting rules. In *AAMAS*, pages 53–60, 2014.

[21] B. Escoffier and V. Paschos. Differential approximation of MinSat, MaxSat and related problems. *EJOR*, 181(2):620–633, 2007.

[22] D. M. Kilgour. Approval balloting for multi-winner elections. In *Handbook on Approval Voting*, chapter 6, pages 105–124. 2010.

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

[24] J. Lang and L. Xia. Voting on combinatorial domains. In *Handbook of Computational Social Choice*. Cambridge University Press, 2016.

[25] J.-F. Laslier and R. Sanver, editors. *Handbook on Approval Voting*. Springer, 2010.

[26] R. LeGrand, E. Markakis, and A. Mehta. Some results on approximating the minimax solution in approval voting. In *AAMAS*, pages 1185–1187, 2007.

[27] M. Li, Q. B. Vo, and R. Kowalczyk. Majority-rule-based preference aggregation on multi-attribute domains with CP-nets. In *AAMAS*, pages 659–666, 2011.

[28] R. Meir, A. Procaccia, J. Rosenschein, and A. Zohar. Complexity of strategic behavior in multi-winner elections. *J. Artif. Intell. Res.*, 33:149–178, 2008.

[29] N. Misra, A. Nabeel, and H. Singh. On the parameterized complexity of minimax approval voting. In *AAMAS*, pages 97–105, 2015.

[30] F. Moti and L. Ami. On covering problems of codes. *Theory of Computing Systems*, 30(2):113–119, 1997.

[31] A. Procaccia, J. Rosenschein, and A. Zohar. On the complexity of achieving proportional representation. *Social Choice and Welfare*, 30(3):353–362, 2008.

[32] F. Rossi, K. Venable, and T. Walsh. mCP nets: representing and reasoning with preferences of multiple agents. In *AAAI*, pages 729–734, 2004.

[33] P. Skowron and P. Faliszewski. Fully proportional representation with approval ballots: approximating the maxcover problem with bounded frequencies in FPT time. In *AAAI*, 2015.

Nathanaël Barrot
Laboratoire d'analyse et modélisation de systèmes pour l'aide à la décision (LAMSADE)
Dauphine University
Paris, France
Email: nathanael.barrot@lamsade.dauphine.fr

Jérôme Lang
CNRS, Laboratoire d'analyse et modélisation de systèmes pour l'aide à la décision (LAMSADE)
Dauphine University
Paris, France
Email: lang@lamsade.dauphine.fr