How Hard Is It to Control Sequential Elections Via the Agenda?

Vincent Conitzer
Department of Computer Science
Duke University
Durham, NC 27708, USA
conitzer@cs.duke.edu

Jérôme Lang
LAMSADE
Université Paris-Dauphine
75775 Paris Cedex 16, France
lang@lamsade.dauphine.fr

Lirong Xia
Department of Computer Science
Duke University
Durham, NC 27708, USA
lxia@cs.duke.edu

Abstract

Voting on multiple related issues is an important and difficult problem. The key difficulty is that the number of alternatives is exponential in the number of issues, and hence it is infeasible for the agents to rank all the alternatives. A simple approach is to vote on the issues one at a time, in sequence; however, a drawback is that the outcome may depend on the order in which the issues are voted upon and decided, which gives the chairperson some control over the outcome of the election because she can strategically determine the order. While this is undeniably a negative feature of sequential voting, in this paper we temper this judgment by showing that the chairperson's control problem is, in most cases, computationally hard.

1 Introduction

In many real-life group decision making problems, the space of alternatives has a multiattribute (or combinatorial) structure. In one example [Brams *et al.*, 1998], the inhabitants of a community have to make a common decision over several related issues of local interest, such as which ones of several public facilities to build. As another example, the members of an association may have to elect a steering committee, composed of a president, a vice-president and a treasurer. In both cases, the space of alternatives has a combinatorial structure: there are multiple issues (aka. *attributes* of the alternatives) to decide on, and voters generally have preferential dependencies among these attributes. For instance, a voter might want a tennis court to be built only if no swimming pool is built.

In classical voting theory, voters submit their preferences as *linear orders* over the set of alternatives, and then a *voting rule* is applied to select one alternative to be the winner[1]. However, when the set of alternatives has a multiattribute structure, the number of alternatives is exponential in the number of attributes, and therefore it is not realistic to ask voters to specify their preferences as explicit linear orders. We can consider the following three ways to address this:

1. Decompose the vote into a set of *parallel*, independent voting problems, one for each attribute. This may lead to catastrophic results as soon as some voters have preferential dependencies among attributes [Lacy and Niou, 2000].

2. Ask voters to report their preferences in some compact representation language, as in [Xia *et al.*, 2008]. While this may be practical in some cases, in the general case we are confronted with two problems: the size of the input to be elicited from the voters (exponentially large in the worst case) and the generally very high computational complexity of computing the outcome of the election.

3. Have the voters vote on each of the attributes in sequence (the approach studied in this paper). That is, we decide the value of the attributes one after the other, using a *local voting rule* for each attribute. This solution has the nice features of being elicitation-friendly and computationally easy (provided the local rules are computationally easy to run). However, sequential voting has two severe drawbacks. First, voters may feel ill at ease voting when their preferences for the current attribute depend on values of other attributes that have not been decided yet. Second, the outcome of the process may depend on the order in which the attributes are voted on.

Some earlier work [Lang, 2007; Xia *et al.*, 2007] avoids these two pitfalls of sequential voting by requiring that there exists some order over the attributes that is consistent with the preferences—that is, a voter's preferences for an attribute never depend on the values of attributes later in the order. In this case, it is natural to vote over the attributes in this order, because then each voter will have a clear preference for each attribute when it is voted on. (While in principle the chairperson could still attempt to control the election by insisting on another order without this property, in practice this would presumably appear very suspect to the voters.) However, assuming the existence of such an order is very restrictive in general: the number of preferences (linear orders over the alternatives) satisfying the restriction is exponentially smaller than the number of possible preferences [Xia *et al.*, 2008].

In this paper, we do not assume that there is such a natural order in which to vote. As mentioned above, this leads to several problems, which we now illustrate with an example.

Example 1 *A joint decision has to be made about whether or not to build a new swimming pool (S or $\bar{S}$) and a new tennis court (T or $\bar{T}$). We suppose that the preferences of voters 1 to 10 are $S\bar{T} \succ \bar{S}\bar{T} \succ \bar{S}\bar{T} \succ ST$, those of voters 11 to 20 are $\bar{S}T \succ S\bar{T} \succ \bar{S}\bar{T} \succ ST$, and those of voters 21 to 30 are $ST \succ S\bar{T} \succ \bar{S}T \succ \bar{S}\bar{T}$.*

The first problem is that, regardless of the order in which

[1] Alternatively, a *voting correspondence* can be applied to select multiple winners. We onl focus on voting rules in this paper.

the issues are decided on, voters 1 to 20 feel ill at ease when asked to report their preference about the first issue. They prefer S to $\bar{S}$ if and only if $\bar{T}$ is chosen, so they would like to know about the tennis court before deciding on the swimming pool. However, deciding on the tennis court first leads to similar issues, because they prefer T to $\bar{T}$ if and only if $\bar{S}$ is chosen. Only voters 21 to 30 can safely vote for S over $\bar{S}$.[2]

The second problem is that in this type of situation, the outcome of the election can depend on the order in which the attributes are decided on. For example, suppose that the swimming pool is decided on first, and more than half of the first 10 voters vote (optimistically) for S; then, S will be chosen over $\bar{S}$. Subsequently, given that they know that the swimming pool will be built, 20 voters will vote against the tennis court, so that the final outcome will be $S\bar{T}$. However, if the tennis court is decided on first, then optimistic voting would result in a final outcome of $\bar{S}T$. As a consequence, if the chairperson (chair) knows enough about the voters' behavior, she will be able to predict the outcome for each order and thus she will be tempted to choose the order $\mathbf{S} > \mathbf{T}$ if she prefers $S\bar{T}$ to $\bar{S}T$, and $\mathbf{T} > \mathbf{S}$ otherwise[3].

The fact that the chair has, in some situations, the ability to influence the outcome of the election by choosing the *agenda* (that is, the order in which the issues are decided on) is a definite drawback of sequential voting. However, we might wonder *how difficult* it is for the chair to control the election in this way. There are various other ways in which a chair might control an election, and the computational complexity of some of those types of control has been considered in several places, starting with [Bartholdi *et al.*, 1992] and later on in [Faliszewski *et al.*, 2007; Hemaspaandra *et al.*, 2007b; 2007a; Meir *et al.*, 2008; Pini *et al.*, 2008]. In the control problems considered in these papers, the chair may add or remove candidates or voters, or fix the competition tree when applying the "cup" rule. However, the control problem we study here is unique because of its multiattribute nature, and, in our opinion, also quite realistic. If we are able to prove that this control problem is computationally hard, this would be an argument in favor of sequential voting, because it mitigates the downside of agenda control.

Before we can study the complexity of the control problem, we first need to formulate it. As we saw in the above example, the behavior of a voter can generally not be determined from her preferences alone. We assume that the chair knows what every voter would do in every situation, where a *situation* consists of a list of decisions made on some of the issues, and another issue which is currently being voted on. In Section 2, we formulate the problem precisely and discuss possible ways of representing this knowledge in a compact way. In Section 3, we prove that in this setting, constructive control (deciding whether there exists an order that makes a given candidate win) is NP-complete, which answers our main question. In Section 4 we show that destructive control (deciding whether there exists an order that ensures a given candidate does not win) is also NP-complete. Lastly, we conclude in Section 5.

[2]Experimental studies suggest that most voters tend to report their preferences optimistically in such situations [Plott and Levine, 1978]; for instance, voters 1–10 would likely report a preference for S over $\bar{S}$.

[3]Incidentally, if the issues are voted on in parallel rather than sequentially, then the chair has no influence over the outcome of the election—but, again under the realistic assumption that voters tend to vote optimistically, the outcome would be ST, which is the worst outcome for two thirds of the electorate!

2 Formulating the problem

In the rest of this paper, there are a set of voters $\{1, \ldots, n\}$ and a set of binary issues $I = \{\mathbf{x}_1, \ldots, \mathbf{x}_p\}$. The notions and results introduced in this paper would easily extend to non-binary issues, but we focus on binary issues so that we do not have to discuss the choice of the local rule for each issue—for binary issues, the majority rule is the obvious choice.

The input of the problem should contain the following information (known by the chair): for every voter i, every issue $\mathbf{x}_j$, and every possible tuple $\vec{z}$ of values assigned to a subset of issues not including $\mathbf{x}_j$, the behavior of the voter (vote for/against $\mathbf{x}_j$). For instance, in Example 1, we assume that the chair knows what every voter would say, were she asked to vote for or against the swimming pool, in each of the following three situations: the swimming pool is the first issue to be decided; we have already decided to build a tennis court; we have already decided not to build a tennis court. Similarly, the chair is assumed to know the voter's behavior when voting on the tennis court, in each of the three possible situations.

Formally, let $Z_i = \{\top, \bot, u\}^{I \setminus \{\mathbf{x}_i\}}$ be the set of all possible situations when asking a voter about $\mathbf{x}_i$. An element $s \in Z_i$, called an *i-situation* (or, for short, a situation), is denoted by listing all variables $\mathbf{x}_j$ in $I \setminus \{\mathbf{x}_i\}$ with their corresponding values: x_j (meaning that $\mathbf{x}_j$ has already been assigned to true ($\top$)) $\overline{x_j}$ ($\mathbf{x}_j$ has already been assigned to false ($\bot$)), or $u(\mathbf{x}_j)$ ($\mathbf{x}_j$ has not been assigned a value yet). For instance, if $I = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \mathbf{x}_4\}$ then the 1-situation $\overline{x_2}u(\mathbf{x}_3)x_4$ means that $\mathbf{x}_2$ has been assigned to $\bot$, $\mathbf{x}_4$ to $\top$, and $\mathbf{x}_3$ has not been assigned yet. The situation where none of these issues has been assigned yet is denoted by $\emptyset$.

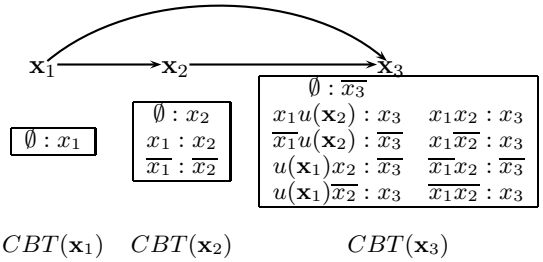
A *voter behavior policy* is a function $\pi : \{(\mathbf{x}_i, \vec{z}) \mid \vec{z} \in Z_i\} \to \{\top, \bot\}$. $\pi(\mathbf{x}_i, \vec{z}) = \top$ (resp., $\bot$) means that in the i-situation $\vec{z}$, when asked to vote for or against x_i, the voter votes for $\mathbf{x}_i = \top$ (resp., $\mathbf{x}_i = \bot$). In Example 1, those among voters 1 to 10 who behave optimistically would have the following behavior policy:

$$\pi(\mathbf{S}, \emptyset)(= \pi(\mathbf{S}, u(T))) = \top \quad \pi(\mathbf{S}, T) = \bot \quad \pi(\mathbf{S}, \bar{T}) = \top$$
$$\pi(\mathbf{T}, \emptyset)(= \pi(\mathbf{T}, u(S))) = \bot \quad \pi(\mathbf{T}, S) = \bot \quad \pi(\mathbf{T}, \bar{S}) = \top$$

We may ask for a *consistency condition*, which we will describe shortly. First, we have to introduce the following notion: let $\mathbf{x}_i$ be an issue and consider two i-situations s, s' satisfying the following condition: there exists an issue $\mathbf{x}_j$ such that (a) s, s' coincide on all issues except j; (b) s assigns $\top$ to $\mathbf{x}_j$ and (c) s' assigns $\bot$ to $\mathbf{x}_j$. Then, s, s' are said to be *$\mathbf{x}_j$-conjugate*, and $s \vee s'$ denotes the i-situation that coincides with s (and s') for all issues other than $\mathbf{x}_j$, and leaves $\mathbf{x}_j$ unassigned. Now, a voter behavior policy π is *consistent* if the following condition is met: for any issue $\mathbf{x}_i$ and any pair of $\mathbf{x}_j$-conjugate i-situations s, s', if $\pi(s) = \pi(s')$ then $\pi(s \vee s') = \pi(s)(= \pi(s'))$. The intuition behind consistency is the following. Suppose we know that a voter will vote for the swimming pool both in the case where it has already been decided that the tennis court will be built ($\pi(S, T) = \top$), and

in the case where it has already been decided that it will not be built ($\pi(S, \bar{T}) = \top$). Then this voter should also vote for the swimming pool in the situation where nothing has been decided about the tennis court ($\pi(S, u(T)) = \top$). While consistency is intuitively a desirable condition, psychological experiments show that it is sometimes violated by individuals [Tversky and Shafir, 1992].

Representing a behavior policy explicitly, by listing all elements of Z_i for every i, would require exponential space (to be precise, exactly $p \cdot 3^{p-1}$ values have to be specified) and is therefore infeasible for all but the smallest p. In most cases, however, a voter's behavior on an issue depends on only a small subset of the other issues—just as in CP-nets [Boutilier *et al.*, 2004], a voter's preference for an issue may depend on only a small subset of the other issues. We define a *conditional behavior net* (CB-net) over a set of binary issues I by a directed graph G over I, and, for every $\mathbf{x}_i \in I$, a *conditional behavior table* (CBT) mapping every element of $\{\top, \bot, u\}^{Par_G(\mathbf{x}_i)}$ (where $Par_G(\mathbf{x}_i)$ denotes the set of parents of $\mathbf{x}_i$ in G), to $\top$ or $\bot$. The main technical difference between a CB-net and a CP-net over propositional variables are that in a CB-net, the conditional table for $\mathbf{x}_i$ specifies a value even if some of the parents of $\mathbf{x}_i$ are unassigned. Here is an example of a CB-net (with $I = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3\}$):



Every CB-net Γ induces a behavior policy π_Γ, defined as follows: for every issue $\mathbf{x}_i$ and every $\vec{z} \in Z_i$, $\pi(\mathbf{x}_i, \vec{z}) = \top$ (resp., $\bot$) if $CBT(\mathbf{x}_i)$ contains an entry $\vec{t} : x_i$ (resp., $\vec{t} : \overline{x}_i$) such that $\vec{t} \subseteq \vec{z}$.

The behavior policy associated with the example CB-net above is consistent; it would not be if, instead of $x_1 u(\mathbf{x}_2) : x_3$, we had $x_1 u(\mathbf{x}_2) : \overline{x_3}$—because then we would have two $\mathbf{x}_2$-conjugate situations, $x_1 x_2$ and $x_1 \overline{x_2}$, such that the voter votes for x_3 in both, and yet votes for $\overline{x_3}$ in situation $x_1 u(\mathbf{x}_2)$.

Let $\vec{\pi} = \langle \pi_1, \ldots, \pi_n \rangle$, where π_i is the voter behavior policy for voter i. Finally, let O be an ordering on I. The outcome associated with O and $\vec{\pi}$, denoted by $Seq(O, \vec{\pi})$, is defined as follows. Without loss of generality, assume $O = \mathbf{x}_1 > \ldots > \mathbf{x}_p$. For the sake of simplicity, assume we have an odd number of voters, and that every local rule is the majority rule[4]. $Seq(O, \vec{\pi}) = (v_1, \ldots, v_p)$ where

• $v_1 = x_1$ (resp., $\overline{x_1}$) if for a majority of voters $j \in \{1, \ldots, n\}$ we have $\pi_j(\mathbf{x}_1, \emptyset) = \top$ (resp., $\bot$);

• for every $2 \leq i \leq p$, $v_i = x_i$ (resp., $\overline{x_i}$) if for a majority of voters $j \in \{1, \ldots, n\}$ we have $\pi_j(\mathbf{x}_i, \vec{z}_{i-1}) = \top$ (resp., $\bot$), where $\vec{z}_{i-1} = (v_1, \ldots, v_{i-1}, u(\mathbf{x}_{i+1}), \ldots, u(\mathbf{x}_p))$.

[4]Recall that the issues are binary, hence the obvious choice of the majority rule, with some tie-breaking mechanism; the assumption that we have an odd number of voters allows us to ignore the tie-breaking mechanism. If issues were not binary, then we would have $r_1, \ldots, r_p$ be local voting rules, one for each issue.

We now define the control problems that we study. We distinguish between *local control*, where the chair tries to determine the outcome of a single issue, and *global control*, where the chair tries to determine the winning alternative (that is, the value of all issues). We also distinguish between *constructive control*, where the chair tries to ensure that a particular alternative or a particular value for an issue wins, and *destructive control*, where the chair tries to ensure that a particular alternative or a particular value for an issue does *not* win. Thus, a *local control instance* is defined by a CB-net for every voter, a distinguished issue $\mathbf{x}_i$, and a value $v \in \{\top, \bot\}$. The chair can constructively (destructively) control the instance if there exists an ordering O such that $(Seq(O, \vec{\pi}))_i = v$ (such that $(Seq(O, \vec{\pi}))_i \neq v$). A *global control instance* is defined by a CB-net for every voter, and a distinguished alternative $\vec{x} \in 2^I$. The chair can constructively (destructively) control the instance if there exists an ordering O such that $Seq(O, \vec{\pi}) = \vec{x}$ (such that $Seq(O, \vec{\pi}) \neq \vec{x}$).

In two of our reductions, we make use of the following result, which states that we can simulate any (consistent or not) weighted CB-net (that is, a CB-net with an integer weight on each entry) whose weights are all even with a collection of consistent CB-nets – thus the result we give below is a kind of McGarvey theorem [McGarvey, 1953] (which states that for every tournament T, there is a profile of which T is the majority graph) for CB-nets. If π is the behavior policy corresponding to CB-net Γ, and $\vec{z} \in \{\top, \bot, u\}^{Par_G(\mathbf{x}_i)}$ is a valuation for the parents of $\mathbf{x}_i$, then let $\Gamma|_{\mathbf{x}_i:\vec{z}} = \pi(\mathbf{x}_i, \vec{z}')$ where $\vec{z}'$ is any valuation for all issues other than $\mathbf{x}_i$ that agrees with $\vec{z}$. That is, $\Gamma|_{\mathbf{x}_i:\vec{z}}$ is the CBT entry for $\mathbf{x}_i$ given parent valuations $\vec{z}$.

Proposition 1 *For any CB-net Γ (with binary issues, consistent or not), in which any issue $\mathbf{x}_i$ and any valuation $\vec{z}$ of the parents of $\mathbf{x}_i$ in the dependency graph of Γ is associated with an even[5] $w_{\vec{z},i}$ (which must be positive if $\Gamma|_{\mathbf{x}_i:\vec{z}} = \top$ and negative if $\Gamma|_{\mathbf{x}_i:\vec{z}} = \bot$), there exists a collection of consistent CB-nets, $\vec{\Gamma} = (\Gamma_1, \ldots, \Gamma_K)$, where K is the maximum sum of the absolute values of the weights associated with an issue (summing over the parents' valuations), such that*

• *for any $j \leq K$, the dependency graph of Γ_j is the same as the dependency graph of Γ;*

• *for any $i \leq p$, any $\vec{z} \in \{\top, \bot, u\}^{Par_G(\mathbf{x}_i)}$, $w_{\vec{z},i} = |\{j \leq K : \Gamma_j|_{\mathbf{x}_i:\vec{z}} = \top\}| - |\{j \leq K : \Gamma_j|_{\mathbf{x}_i:\vec{z}} = \bot\}|$. That is, the weights in the weighted CB-net Γ correspond exactly to the weights of the majority relation for $(\Gamma_1, \ldots, \Gamma_K)$.*

Proof of Proposition 1: For any $i \leq p$, we let W_i be the sum of the absolute values of the weights over all valuations of the parents of $\mathbf{x}_i$ in the dependency graph of Γ. Because the ordering $\mathbf{x}_1, \ldots, \mathbf{x}_p$ of the issues is arbitrary (the CB-net may be cyclic), we can without loss of generality assume that $K = W_p$, that is, for any $i \leq p$, $W_p \geq W_i$. Let $\mathbf{x}_1, \ldots, \mathbf{x}_k$ be the parents of $\mathbf{x}_p$. We next show how to construct $CBT(\mathbf{x}_p)$ for each Γ_j in $\vec{\Gamma}$. For any $1 \leq i \leq p-1$, $CBT(\mathbf{x}_i)$ for each Γ_j can be constructed similarly. We now prove a lemma whose purpose is to construct a CBT such that the consistency condition does not cause us problems.

[5]Alternatively, all the weights can be odd.

Lemma 1 *Let $\mathbf{x}_1, \ldots, \mathbf{x}_k$ be the parents of $\mathbf{x}_p$. There exists a CBT for $\mathbf{x}_p$ such that for any $i \leq k$, any pair of $\mathbf{x}_i$-conjugate p-situations s, s', we have that $\mathbf{x}_p$ is assigned $\top$ in exactly one of $\{s, s'\}$, and $\bot$ in the other.*

Proof of Lemma 1: The "parity CB-net," which returns $\top$ when an even number of parents are set to $\top$, and $\bot$ otherwise, satisfies this condition. $\square$

For any valuation s' of $Par_G(\mathbf{x}_p)$, let $f_p(s')$ be the value that the CBT from Lemma 1 returns. For any valuation s of $Par_G(\mathbf{x}_p)$, we assume $w_{s,p} > 0$ (the case where $w_{s,p} < 0$ is similar). Let $I(s)$ be the set of issues that, under s, are assigned a value in $\{\top, \bot\}$ (that is, not u). We define two CB-nets for every valuation s of $Par_G(\mathbf{x}_p)$, namely, $CBT_s^1(\mathbf{x}_p)$ and $CBT_s^2(\mathbf{x}_p)$. Let $CBT_s^1(\mathbf{x}_p)(s')$ be $\top$ if $I(s') \subseteq I(s)$, and let $CBT_s^1(\mathbf{x}_p)(s')$ be $f_p(s')$ otherwise. $CBT_s^2(\mathbf{x}_p)$ is *almost* the opposite: let $CBT_s^2(\mathbf{x}_p)(s')$ be $\bot$ if $I(s') \subseteq I(s)$ *and* $s' \neq s$, let it be $\top$ if $s' = s$, and let it be $\neg f_p(s')$ otherwise. Next, we show (1) that these CBTs do not violate consistency, and (2) that, if we aggregate some combination of these CBTs, then we get Γ's weighted CBT for $\mathbf{x}_p$.

For (1): for any $i \leq k$, any pair of i-conjugate valuations s_1, s_2 of $Par_G(\mathbf{x}_p)$, if $I(s_1)(= I(s_2)) \nsubseteq I(s)$, then consistency is maintained in both $CBT_s^1(\mathbf{x}_p)$ and $CBT_s^2(\mathbf{x}_p)$, because $\{f_p(s_1), f_p(s_2)\} = \{\neg f_p(s_1), \neg f_p(s_2)\} = \{\top, \bot\}$, so consistency does not impose any constraint. If $I(s_1)(= I(s_2)) \subseteq I(s)$, and neither s_1 nor s_2 is equal to s, then $I(s_1 \vee s_2) \subseteq I(s)$. Because of this, $CBT_s^1(\mathbf{x}_p)(s_1) = CBT_s^1(\mathbf{x}_p)(s_2) = CBT_s^1(\mathbf{x}_p)(s_1 \vee s_2)$ and $CBT_s^2(\mathbf{x}_p)(s_1) = CBT_s^2(\mathbf{x}_p)(s_2) = CBT_s^2(\mathbf{x}_p)(s_1 \vee s_2)$, so consistency is also maintained in this case. Finally, if one of s_1 and s_2 is equal to s, then, for $CBT_s^1(\mathbf{x}_p)$, the situation is the same as in the previous case; for $CBT_s^2(\mathbf{x}_p)$, we have $CBT_s^2(\mathbf{x}_p)(s_1) \neq CBT_s^2(\mathbf{x}_p)(s_2)$, so consistency does not impose any constraint.

For (2): we have $CBT_s^1(\mathbf{x}_p)(s) = CBT_s^2(\mathbf{x}_p)(s) = \top$, and for every $s' \neq s$, we have $CBT_s^1(\mathbf{x}_p)(s') \neq CBT_s^2(\mathbf{x}_p)(s')$. Thus, they cancel each other out except on s, where they have a combined weight of 2 on s. Hence, if we take $\frac{w_{s,p}}{2}$ copies of each, we have a weight of $w_{s,p}$ on s (and 0 everywhere else). If we do this for every value s of $\mathbf{x}_p$'s parents, then we obtain the weighted CBT of Γ for $\mathbf{x}_p$. The total number of CBTs that we create in this process is K. We use the same approach for the issues other than $\mathbf{x}_p$, and each of them requires at most K CBTs. Moreover, we can take a CBT for each issue and combine them into a single CB-net, so that the total number of CB-nets we create is K (for the issues that require fewer than K CBTs, we can create some pairs of CBTs that cancel out exactly with each other). $\square$

Of course, in Proposition 1, all the weights $w_{s,i}$ may have the same absolute value, *i.e.*, $w_{s,i} = 2$ or $w_{s,i} = -2$, so that Γ is effectively an unweighted CB-net. Hence, we can simulate a single inconsistent CB-net by a collection of consistent CB-nets. In this case, the number of consistent CB-nets needed is at most 2^{k+1}, where k is the maximum in-degree of vertices in the dependency graph of Γ. This number is polynomial when k is bounded, as it will be in our reductions.

3 Constructive control

We now investigate the complexity of constructive control. It turns out that both the local and global versions of constructive control are computationally hard.

Proposition 2 *Local constructive control is* NP*-complete.*

Proposition 3 *Global constructive control is* NP*-complete.*

In both cases, NP-hardness holds even if every issue is binary and the local rules are majority rules, and either of the following conditions holds: there is only one voter *or* all the CB-nets are consistent and have the same dependency graphs. While neither of these hardness results directly implies the other, we use a single reduction to prove both results.

Proof sketch of Propositions 2 and 3. In both cases, membership in NP is straightforward. As for hardness, we use a reduction from the restriction of HAMILTONIAN CYCLE to graphs where each node has degree at most 3; this restriction is still NP-complete [Garey *et al.*, 1976]. Let $G = (V, E)$ be an undirected graph, where $V = \{0, \ldots, n\}$. We denote edges by ij (where $i, j \in V$, $i \neq j$) rather than $\{i, j\}$; ij and ji represent the same edge. We first define the CB-nets used in our control problem. Rather than listing the individual CB-nets explicitly, we give the majority CB-net, which can correspond either to a single not-necessarily-consistent CB-net, or, by Proposition 1, to a collection of consistent CB-nets. In the construction we assume the degree of each vertex is 3 for simplicity; it is simple to adapt the construction to the case where some vertices have degree less than 3.

1. there is an issue ij for every $\{i, j\} \in E$ (with $i < j$); an issue $done(i)$ for every $i = 0, \ldots, n$; and an issue $done$.

2. let $0i \in E$; $0j$, $0k$ are the other two edges with 0 as extremity; and il, im are the other two edges with i as extremity. Then the parents of $0i$ in the dependency graph are $0j$, $0k$, il, im and $done$, and the table for issue $0i$ is:

$\emptyset$: $0i$	$0j\ u(0k)\ il\ u(im)\ u(done)$: $0i$
$0j\ u(0k)\ u(il)\ im\ u(done)$: $0i$	$u(0j)\ 0k\ il\ u(im)\ u(done)$: $0i$
$u(0j)\ 0k\ u(il)\ im\ u(done)$: $0i$	$\times \times \times \times\ done$: $0i$
all other situations : $\overline{0i}$	

The $\times$ notation refers to any possible status ($\top$, $\bot$ or unassigned) of the issues $0j$, $0k$, il and im, so, it is a shorthand for 16 conditional behaviors. "All other situations" is an abbreviation referring to all tuples not explicitly mentioned in the table. Using these abbreviations or not does not change anything regarding the polynomial size of the reduction, because each issue has at most three parents in the dependency graph.

3. let $ij \in E$, $i \neq 0$, $j \neq 0$; let ik, il be the other two edges with i as extremity, and jm, jq the other two edges with j as extremity. Then the parents of ij in the dependency graph are ik, il, jm, jq and $done$, and the table for issue ij is

$ik\ u(il)\ u(jm)\ u(jq)\ u(done)$: ij
$u(ik)\ il\ u(jm)\ u(jq)\ u(done)$: ij
$u(ik)\ u(il)\ jm\ u(jq)\ u(done)$: ij
$u(ik)\ u(il)\ u(jm)\ jq\ u(done)$: ij
$\times \times \times \times\ done$: ij
all other situations : $\overline{ij}$

4. let $0i$, $0j$, $0k$ be the three edges in G with 0 as extremity; then the parents of $done(0)$ in the dependency graph are $0i$, $0j$, $0k$ and the table for $done(0)$ is

$0i\ \ 0j\ \ u(0k) : done(0)$	$0i\ \ u(0j)\ \ 0k\ \ : done(0)$
$u(0i)\ \ 0j\ \ 0k : done(0)$	all other situations : $\overline{done(0)}$

5. for $1 \leq i \leq n$, let ij, ik, il be the three edges in G with i as extremity; then, the parents of $done(i)$ in the dependency graph are ij, ik, il, $done(i-1)$, and the table for $done(i)$ is

$ij\ \ ik\ \ u(il)\ \ done(i-1) : done(i)$	$ij\ \ u(ik)\ \ il\ \ done(i-1) : done(i)$
$u(ij)\ \ ik\ \ il\ \ done(i-1) : done(i)$	all other situations : $\overline{done(i)}$

6. the only parent of $done$ is $done(n)$, and its table is

$done(n) :\ \ done$	all other situations $\overline{done}$

Finally, the goal of the chair depends on the version (local or global) of the constructive control problem. In the global version (Proposition 3), it is to get the outcome $\vec{x}^*$ where every issue is assigned to $\top$. In the local version (Proposition 2), it is to get the issue $done$ assigned to $\top$.

Note that this CB-net is not consistent. Take issue ij ($i, j \neq 0$). In situation (ik, il, $u(jm)$, $u(jq)$), the decision is $\overline{ij}$. In situation (ik, $\overline{il}$, $u(jm)$, $u(jq)$), it is $\overline{ij}$ too. But, in situation (ik, $u(il)$, $u(jm)$, $u(jq)$), the decision is ij. However, we note that an inconsistent CB can be obtained as the majority CB-net of a set of consistent CB-nets (Proposition 1).

The proof goes along the following lemmas.

Lemma 2 *If there exists a Hamiltonian cycle, then there exists an order leading to $\vec{x}^*$.*

Lemma 3 *For any partial assignment $\vec{z}$ of the issues, let $S(\vec{z})$ be the following subgraph of G: $ij \in S(\vec{z})$ if and only if ij has been assigned $\top$ under $\vec{z}$. Then, as long as $done$ has not been assigned to $\top$, the following properties are true:*

1. for every vertex i, $S(\vec{z})$ contains zero, one or two edges adjacent to i (never three);

2. if $S(\vec{z})$ contains no edge adjacent to 0, then $S(\vec{z})$ is empty;

3. if $S(\vec{z})$ contains one edge adjacent to 0, then there is exactly one $i \neq 0$ such that $S(\vec{z})$ contains one edge adjacent to i, and $S(\vec{z})$ is a single path going from 0 to i.

4. if $S(\vec{z})$ contains two edges adjacent to 0, then for every i, $S(\vec{z})$ contains zero or two edges adjacent to i, and $S(\vec{z})$ is a cycle including exactly the vertices i such that $S(\vec{z})$ contains two edges adjacent to i.

Lemma 4 *$done(0)$ can be assigned to $\top$ if and only if there are exactly two issues $0j$, $0k$ that have been assigned to $\top$.*

Lemma 5 *For any $1 \leq i \leq n$, $done(i)$ can be assigned to $\top$ if and only if $done(i-1)$ has been assigned to $\top$, and there are exactly two issues ij, ik that have been assigned to $\top$.*

Lemma 6 *$done$ can be assigned to $\top$ if and only if there is a Hamiltonian cycle.*

We only sketch the proofs of the lemmas. For Lemma 2, suppose w.l.o.g. that the Hamiltonian cycle is $(0, 1, 2, \ldots, n, 0)$. Consider the following order on issues: 01, 12, ..., $n-1\ n$, $0n$, $done(0)$, $done(1)$, ..., $done(n)$, $done$, and then all remaining issues in any order afterwards. The answer given by the majority CB-net at each step is $\top$. The four points of Lemma 3 follow from the conditional behavior tables (points 3 and 4 are proven by induction on the number of issues assigned to $\top$). Lemmas 4 and 5 follow directly from the behavior tables for $done(i)$, $i = 0, \ldots, n$. For Lemma 6, $done$ is assigned to $\top$ *iff* every $done(i)$ has been assigned to $\top$, which, by Lemma 5, implies that for every i, exactly two issues ij have been assigned to $\top$. Let $\vec{z}$ be the current assignment. From point 4 of Lemma 3, $S(\vec{z})$ is a cycle containing every vertex of G, therefore this subgraph is a Hamiltonian cycle. Proposition 2 follows from Lemma 6, and Proposition 3 from Lemmas 2 and 6. *End of proof of Prop. 2, 3.* □

We can also show an inapproximability result for global control: unless P = NP, there does not exist a polynomial-time algorithm that, whenever the global control instance has a solution, returns a solution that disagrees with the desired outcome on only a small number of issues. This result is easily obtained by modifying the reduction used in the proof of Proposition 3 by adding an arbitrary large number of dummy issues that can be assigned to $\top$ *iff* $done$ is assigned to $\top$.

We finish this section by giving a subclass of problems where constructive control (both local and global) is easy.

Proposition 4 *If the voters' CB-nets share the same dependency graph G, and every node in G has at most one parent. then constructive control, both local and global, is in* P.

Proof sketch. For the global version, assume w.l.o.g. that $\vec{x}^* = x_1 \ldots x_p$. Let N be the majority CB-net obtained from the individual CB-nets. For each issue $\mathbf{x}_i$, we create a simple precedence constraint between $\mathbf{x}_i$ and its parent $\mathbf{x}_j$ in G. For instance, if N contains the entry $x_j : x_i, u(\mathbf{x}_j) : \overline{x_i}$, then the control of $\mathbf{x}_i$ will succeed *iff* $\mathbf{x}_i$ is decided after $\mathbf{x}_j$. It is then easy to check if there exists an order on the issues such that all of these p constraints can be satisfied. Local constructive control can be solved in a similar way. □

4 Destructive control

Finally, we consider destructive control. For the local version, NP-completeness is a straightforward consequence of Proposition 2: being able to make sure that $\mathbf{x}_i$ is assigned to $\top$ is the same thing as being able to avoid that $\mathbf{x}_i$ is assigned to $\bot$. This trivially gives us:

Proposition 5 *Local destructive control is* NP*-complete.*

However, we cannot use such an argument for the global version; it requires a separate reduction.[6]

Proposition 6 *Global destructive control is* NP*-complete.*

NP-hardness holds even if every issue is binary and the local rules are majority rules, and all the CB-nets are consistent.

Proof sketch. Membership in NP is straightforward. Hardness is by reduction from the NP-complete restriction of EXACT COVER BY 3-SETS (X3C) where no element occurs in more than 3 subsets [Garey and Johnson, 1979]: given $X = \{v_1, \ldots, v_q\}, \mathcal{S} = \{S_1, \ldots, S_t\}$, where for every $i \leq t$, $S_i \subseteq X$ and $|S_i| = 3$, and for all $j \leq q$, $|\{S_i \in S : v_j \in S_i\}| \leq 3$, decide whether there exists $\mathcal{S}' \subseteq \mathcal{S}$ such that $|\mathcal{S}'| = \frac{q}{3}$, and $\bigcup_{S_i \in S'} S_i = X$.

[6]Conversely, Proposition 6 also does not seem to directly imply Proposition 3. In settings with a small number of alternatives, constructive control cannot be (significantly) easier than destructive control, because one approach for destructive control is simply to try constructive control on every other alternative. However, this argument does not work in multiattribute settings with exponentially many alternatives.

For any instance of this restricted version of X3C, we construct a global destructive control instance as follows. Let there be $t+1$ issues, $\mathbf{x}, \mathbf{x}_1, \ldots, \mathbf{x}_t$, and $2q+1$ voters, $\pi_1, \ldots, \pi_{2q+1}$, such that: (1) for any $j \leq t$ and $i \leq q$, $\mathbf{x}_j$ is a parent of $\mathbf{x}$ in the dependency graph of π_i *iff* $v_i \in S_j$, and voter i vote for $\mathbf{x}$ to be $\top$ *iff* exactly one of the parents of $\mathbf{x}$ has been assigned to $\top$; (2) for any $j \leq t$ and any $i \in \{1, \ldots, 2q+1\}$, voter i votes for $\mathbf{x}_j$ to be $\top$; (3) for any $i \in \{q+1, \ldots, 2q\}$, voter i unconditionally votes for $\mathbf{x}$ to be $\bot$; (4) voter $2q+1$ unconditionally votes for $\mathbf{x}$ to be $\top$. Lastly, the chair wants to get an outcome different from $\overline{x}x_1 \ldots x_t$.

We note first that every $\mathbf{x}_i$ will unavoidably be assigned to $\top$. Moreover, $\mathbf{x}$ will be assigned to $\top$ if and only if every voter $i \leq q$ votes for $\mathbf{x}$. Suppose there exists an exact 3-cover $\mathcal{S}'$. Without loss of generality, $\mathcal{S}' = \{S_1, \ldots, S_{\frac{q}{3}}\}$. Then, use the order $\mathbf{x}_1 < \ldots < \mathbf{x}_{\frac{q}{3}} < \mathbf{x} < \mathbf{x}_{\frac{q}{3}+1} < \ldots < \mathbf{x}_t$. For each $i \leq q$, v_i is in exactly one element of $\mathcal{S}'$, therefore every voter $i \leq q$ will vote for $\mathbf{x}$ and $\mathbf{x}$ will be assigned to $\bot$.

Conversely, suppose the chair can avoid $\overline{x}x_1 \ldots x_t$, *i.e.*, can ensure that $\mathbf{x}$ will be assigned to $\bot$, which happens only if every voter $i \leq q$ votes for $\mathbf{x}$. Without loss of generality, let the voting order be $\mathbf{x}_1 \ldots, \mathbf{x}_l, \mathbf{x}, \mathbf{x}_{l+1}, \ldots, \mathbf{x}_t$. $\mathcal{S}' = \{S_1, \ldots, S_l\}$ is an exact cover of $\mathcal{S}$, because the fact that voter i votes for $\mathbf{x}$ implies that v_i is exactly in one of the S_j. □

Interestingly, destructive control becomes easy when all voters' CB-nets have the same dependency graphs.

Proposition 7 *Assume all CB-nets share the same directed graph G. Then global destructive control is in* P.

Proof sketch. Let $\vec{x} = (x_1, \ldots, x_n)$. The proof is based on the following fact: if all CB-nets share the same directed graph G, then there exists an order O such that $Seq(O, \vec{\pi}) \neq \vec{x}$ *iff* there exists an issue $\mathbf{x}_i$ and a $K \subseteq Par_G(\mathbf{x}_i)$ such that a majority of voters vote for $\overline{x_i}$ in the situation $\{x_j \mid \mathbf{x}_j \in K\} \cup \{u(\mathbf{x}_j) \mid \mathbf{x}_j \in Par_G(i) \setminus K\}$. Since the size of the CB-nets is exponential in the maximum indegree of G, the number of subsets to check is polynomial in the size of the input, hence the result. □

5 Conclusion

In this paper we have shown that although sequential voting is susceptible to control via the order in which the issues are decided on, this control problem is NP-complete in most cases (for all the variants of this type of control – local or global, constructive or destructive), which is a positive argument for using sequential voting after all. We note that the sparser the voters' dependency graphs, the easier the control problem is, but also, the less likely it is that a successful control exists (in the extreme case where voters have separable preferences, therefore unconditional CB-nets, the chair has no power over the outcome – therefore the control problem is trivial).

Of course, these results are worst-case results, and it would be nice to have results about the *frequency of easy control*, along the same lines as some recent results in the case of manipulation. We can also consider various generalizations of the sequential voting framework, including allowing some issues to be decided in parallel, or allowing for the chair to have some uncertainty about voters' preferences.

Acknowledgements

We thank anonymous reviewers for helpful discussions and comments. Jérôme Lang is supported by the ANR project ANR-05-BLAN-0384 "Preference Handling and Aggregation in Combinatorial Domains." Vincent Conitzer and Lirong Xia are supported by the National Science Foundation under award number IIS-0812113. Vincent Conitzer is supported by an Alfred P. Sloan Research Fellowship, and Lirong Xia is supported by a James B. Duke Fellowship.

References

[Bartholdi *et al.*, 1992] J.J. Bartholdi, C.A. Tovey, and M.A. Trick. How hard is it to control an election? *Mathematical and Computer Modelling*, 16(27–40), 1992.

[Boutilier *et al.*, 2004] C. Boutilier, R. Brafman, C. Domshlak, H. Hoos, and D. Poole. CP-nets: a tool for representing and reasoning with conditional ceteris paribus statements. *JAIR*, 21:135–191, 2004.

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

[Faliszewski *et al.*, 2007] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Llull and copeland voting broadly resist bribery and control. In *Proc. AAAI-07*, 724–730.

[Garey and Johnson, 1979] M. Garey and D. Johnson. *Computers and Intractability*. W. H. Freeman and Company, 1979.

[Garey *et al.*, 1976] M. Garey, D. Johnson, and R. Tarjan. The planar hamiltonian circuit problem is NP-complete. *SIAM J. Comput.*, 5(4):704–714, 1976.

[Hemaspaandra *et al.*, 2007a] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Anyone but him: The complexity of precluding an alternative. *AIJ*, 171(5-6):255–285, 2007.

[Hemaspaandra *et al.*, 2007b] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Hybrid elections broaden complexity-theoretic resistance to control. In *Proc. IJCAI-07*, 1308–1314.

[Lacy and Niou, 2000] D. Lacy and E. Niou. A problem with referenda. *Journal of Theoretical Politics*, 12(1):5–31, 2000.

[Lang, 2007] J. Lang. Vote and aggregation in combinatorial domains with structured preferences. In *Proc. IJCAI-07*, 1366–1371.

[McGarvey, 1953] David C. McGarvey. A theorem on the construction of voting paradoxes. *Econometrica*, 21(4):608–610, 1953.

[Meir *et al.*, 2008] R. Meir, A. Procaccia, J. Rosenschein, and A. Zohar. The complexity of strategic behavior in multi-winner elections. *JAIR*, 33:149–178, 2008.

[Pini *et al.*, 2008] M. Pini, F. Rossi, B. Venable, and T. Walsh. Dealing with incomplete agents' preferences and an uncertain agenda in group decision making via sequential majority voting. In *Proc. KR-08*, 571–578.

[Plott and Levine, 1978] C. Plott and M. Levine. A model of agenda influence on committee decisions. *The American Economic Review*, 68(1):146–160, 1978.

[Tversky and Shafir, 1992] A. Tversky and E. Shafir. The disjunction effect in choice under uncertainty. *Psychological Science*, 3:305–309, 1992.

[Xia *et al.*, 2007] L. Xia, J. Lang, and M. Ying. Sequential voting rules and multiple elections paradoxes. In *Proc. TARK-07*, 279–288.

[Xia *et al.*, 2008] L. Xia, V. Conitzer, and J. Lang. Voting on multi-attribute domains with cyclic preferential dependencies. In *Proc. AAAI-08*, 202-207.