Strategic Considerations in the Design of Committees

Edith Elkind
School of Physical and Mathematical Sciences
Nanyang Technological University, Singapore
eelkind@ntu.edu.sg

Dmitrii Pasechnik
School of Physical and Mathematical Sciences
Nanyang Technological University, Singapore
dima@ntu.edu.sg

Michael Wooldridge
Department of Computer Science
University of Oxford, UK
mjw@cs.ox.ac.uk

ABSTRACT

We study settings in which a central authority must appoint a number of committees, where each committee is tasked with making a specific decision via a given voting rule. Each voter has their own individual preferences, and the center desires the decisions to be made in a certain way. The overall problem is whether the center can design the committees so that if the committee members then vote according to their preferences, the decisions will be made according to the desires of the center. After motivating and formally defining this problem, we investigate cases where this problem can be solved in polynomial time, and highlight cases where the problem is intractable. We consider a range of possible voting rules. We conclude with some possible extensions to the model and future work.

Categories and Subject Descriptors

F.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity;
I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent Systems*

Keywords

Computational Social Choice, Delegation

1. INTRODUCTION

Decision making by committees is a fundamental part of the working life of many organizations. In some organisations—universities are a very obvious example—committees are indeed the primary mechanisms for decision making. Committees also form a key component of decision making in governments (witness e.g., the committees of the US Congress and House of Representatives). Our aim in the present paper is to study committee decision making *from the point of view of a committee designer*. The basic problem we consider is as follows. A central authority (the designer) must appoint $k \geq 1$ committees, where each committee is tasked with making a specific decision via a given voting rule. Each voter has their own individual preferences, and the center desires the decisions to be made in a certain way. The overall problem is whether the center can design the committees so that if the committee members then vote according to their preferences, the decisions will be made according to the desires of the center. Our interest in this problem is not Machiavellian: an individual charged with forming committees in an organisation such as a university may well invest a great deal of energy ensuring that the committees are composed in such a way as to lead to coherent collections of decisions.

Appears in: *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013), Ito, Jonker, Gini, and Shehory (eds.), May, 6–10, 2013, Saint Paul, Minnesota, USA.*

Our paper is directly motivated by recent work of Kraus and Wooldridge [13], which deals with the problem of delegating decisions to individuals in Boolean games (see, e.g., [12, 3, 7] for background on Boolean games). However, our work differs from that of Kraus and Wooldridge in several important aspects. First, in Boolean games agents have to set values of Boolean variables they control rather than rank alternatives. Second, in the model of [13] each decision is delegated to a single agent rather than a committee. Finally, in a Boolean game, when deciding how to set his variable(s), an agent has to predict how the variables controlled by the other agents will be set, and therefore to describe the outcome of the decision-making process one has to engage in equilibrium analysis (and, in particular, deal with the issue of multiple equilibria); in contrast, in our model the voters are assumed to be non-strategic. The similarities and differences between the two models point to several directions for future work (see Section 8).

On a more technical level, our problem can be viewed as a generalization of (a variant of) the the well-studied problem of *constructive control by deleting voters (CCDV)* [2]. In this problem, we are given an election (i.e., a set of candidates, a list of voters together with their preferences over the candidates, and a voting rule), a candidate p and a "budget" B, and we are asked if p can be made the election winner by removing at most B voters. Now, deleting B voters is equivalent to selecting $n - B$ voters to serve on the committee, so CCDV can be viewed a variant of our problem where $k = 1$ (i.e., there is just one issue) and there is a lower bound on the committee size. The complexity of CCDV for various voting rules is quite well understood, and it turns out that many existing hardness results for CCDV can be adapted to our setting, implying that for several voting rules our problem is computationally hard even for $k = 1$ (see Section 6). However, easiness results for CCDV do not necessarily translate into easiness results for our problem, as having to deal with multiple committees provides an additional level of complexity: indeed, we will show that for 2-Approval our problem is NP-hard even for $k = 2$, even though 2-Approval-CCDV is known to be in P [15]. Another closely related problem is winner determination under *lot-based voting rules* [18]. Such rules proceed by first selecting a subset of voters of size s and then applying a given voting rule to the preferences of the voters in this subset. Clearly, deciding if a given candidate has a positive chance of being elected under this procedure is equivalent to our problem with $k = 1$. Walsh and Xia [18] show that winner determination under lot-based voting rules is often NP-hard; we discuss their results in more detail in Section 6.

2. THE MODEL

We are given a set of *issues* $\mathcal{I} = \{1, \ldots, k\}$ and a set of *voters* $N = \{1, \ldots, n\}$. Each issue $j \in \mathcal{I}$ is associated with a set of *alternatives* A^j; we can assume without loss of generality that the size of this set is the same for all issues, i.e., we have $|A^j| = m$ for each $j \in \mathcal{I}$. We denote the elements of A^j by $a_1^j, \ldots, a_m^j$. We say that the issues in $\mathcal{I}$ are *binary* if $m = 2$; in this case we assume that $A^j = \{0^j, 1^j\}$ for each $j \in \mathcal{I}$. For each issue $j \in \mathcal{I}$ each voter $i \in N$ has *preferences* over the set of alternatives for this issue, which are represented by a total order R_i^j over A^j. An *outcome* is a vector of alternatives, one for each issue, i.e., an element of the Cartesian product $\mathcal{A} = A^1 \times \cdots \times A^k$.

The voters are allocated into k *committees* $S^1, \ldots, S^k$; the j-th committee is responsible for making the decision on issue j, i.e., it has to select an alternative from A^j. For ease of presentation, we assume that every voter can serve on every committee, but no voter can serve on two committees simultaneously; in Section 7 we show that both of these constraints can be relaxed. The committee sizes are fixed in advance: we are given a vector $(s^1 \ldots s^k) \in \mathbb{N}^k$, and it is required that $|S^j| = s^j$ for each $j \in \mathcal{I}$. We set $s = \max_{j \in \mathcal{I}} s^j$. We assume that $s^1 + \cdots + s^k \leq n$, and the inequality may be strict, i.e., there may be voters that are not assigned to any of the committees.

Once a committee S^j is formed, its members sincerely report their preferences over A^j. An alternative in A^j is then selected by applying a *voting rule* $\mathcal{F}$, i.e., a mapping that for every set of committee members and a list of their preferences over A^j outputs a unique alternative in A^j. Note that we require voting rules to be defined for an arbitrary number of voters; while it is more standard to keep the number of voters fixed, the current definition is more convenient for our purposes, as committee sizes may vary.

In this paper, we will mostly focus on a family of voting rules known as r-Approval, $r \geq 1$. Under r-Approval an alternative receives one point from each voter that ranks it in top r positions, and the winner is the alternative with the highest number of points, with ties broken according to a fixed lexicographic ordering over the alternatives. That is, if $T \subseteq A^j$ is the set of alternatives for issue j that received the highest number of r-Approval points, then the winner is the alternative a_t^j that satisfies $a_t^j \in T$, $a_\ell^j \notin T$ for every $\ell < t$. The 1-Approval rule is also known as Plurality, and the $(m-1)$-Approval rule (where m is the number of alternatives) is known as Veto. A number of other voting rules are often considered in the literature and could be applied to our setting; these include, for instance, Borda, Copeland, Maximin, and Bucklin, to name a few. For each of these rules the problems considered in this paper can be easily shown to be NP-hard even for $k = 1$, by modifying the known hardness proofs for the closely related CCDV problem. We omit the formal definitions of these rules due to space constraints (an interested reader is referred to, e.g., [1]); we will, however, briefly discuss the corresponding hardness results in Section 6.

We are interested in the situation where the assignment of voters to the committees is performed by a self-interested *central authority* (or, *center*). In Sections 3–6 we assume that the center has a specific outcome $\mathbf{a} = (a_{i_1}^1, \ldots, a_{i_k}^k) \in \mathcal{A}$ in mind, and his goal is to assign voters to committees so that for each $j \in \mathcal{I}$ the voting rule $\mathcal{F}$ outputs $a_{i_j}^j$ when applied to the preferences of the voters in S^j; we consider more general models of the center's preferences in Section 7. Thus, the center's decision problem, which will be the main focus of this paper, can be formalized as follows.

DEFINITION 2.1. *Given a voting rule $\mathcal{F}$, an instance of the $\mathcal{F}$-*DELEGATION *problem is given by a set of issues $\mathcal{I} = \{1, \ldots, k\}$, a list of k committee sizes $s^1, \ldots, s^k$, a set of voters $N = \{1, \ldots, n\}$, a set of alternatives A^j, $|A^j| = m$, for each issue $j \in \mathcal{I}$, an outcome $\mathbf{a} = (a_{i_1}^1, \ldots, a_{i_k}^k) \in A^1 \times \cdots \times A^k$, and, for each issue $j \in \mathcal{I}$ and each voter $i \in N$, a preference order R_i^j over A^j. It is a "yes"-instance if there exists a subset of voters $N' \subseteq N$ and a mapping $\lambda : N' \to \mathcal{I}$ such that for each $j \in \mathcal{I}$ the set $S^j = \lambda^{-1}(j)$ satisfies $|S^j| = s^j$ and, furthermore, the voting rule $\mathcal{F}$ applied to the preferences of voters in S^j outputs $a_{i_j}^j$; otherwise it is a "no"-instance.*

The description of our problem involves several parameters: the number of voters n, the number of issues k, the number of alternatives for each issue m, and the committee size s. Unfortunately, even for very simple voting rules, such as Plurality and Veto, we were not able to find a general efficient procedure that solves DELEGATION for arbitrary values of these parameters. Thus, in what follows we describe algorithms that run in polynomial time when one or more of these parameters can be assumed to be small; while some of these algorithms work for arbitrary polynomial-time computable voting rules, others are designed for specific voting rules, such as Plurality, Veto, or 2-Approval. We will also prove that for sufficiently complex voting rules our problem is computationally hard, even for restricted values of the parameters.

3. GENERAL CASE: EASINESS RESULTS

We first observe that $\mathcal{F}$-DELEGATION is easy for any polynomial-time computable voting rule $\mathcal{F}$ if n is bounded by a constant. Indeed, we can enumerate all possible assignments of voters to committees (the number of such assignments is bounded by $(k+1)^n$, which is polynomial in the input size if n is bounded by a constant), and, for every such assignment, check if it results in the center's preferred outcome. A similar argument applies if both s and k are bounded by a constant: in this case, we have at most sk slots to fill, and there are at most n ways of filling each slot, so we have n^{sk} assignments to consider. Thus, we obtain the following results.

PROPOSITION 3.1. *Let $\mathcal{F}$ be a polynomial-time computable voting rule. Then $\mathcal{F}$-*DELEGATION *admits (a) an algorithm that runs in time $(k+1)^n \cdot \mathrm{poly}(n, m, k, s)$; (b) an algorithm that runs in time $n^{sk} \cdot \mathrm{poly}(n, m, k, s)$.*

Another easy case is that of binary domain, i.e., $m = 2$. Recall that a voting rule is said to be *anonymous* if its output is uniquely determined by the number of voters who submit each preference ranking (rather than their identities); further, a voting rule over $\{0, 1\}$ is said to be *monotone* if its output cannot change from 1 to 0 when some voter's preferred alternative changes from 0 to 1 (see [1] for formal definitions). We will now show $\mathcal{F}$-DELEGATION with $m = 2$ is in P for any polynomial-time computable voting rule over $\{0, 1\}$ that is anonymous and monotone.

Our algorithm for this problem, as well as many other algorithms in this paper, is based on reducing our problem to that of finding a *feasible circulation*. Recall that an instance of the circulation problem is given by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, and, for each arc $(v, w) \in \mathcal{E}$, an upper bound $u(v, w)$ and a lower bound $\ell(v, w)$ on the flow from v to w. A feasible circulation is a collection of values $\{f(v, w)\}_{(v,w) \in \mathcal{E}}$ that satisfies the flow conservation constraints and the capacity constraints, i.e., $\sum_{(z,v) \in \mathcal{E}} f(z, v) = \sum_{(v,w) \in \mathcal{E}} f(v, w)$ for each $v \in \mathcal{V}$ and $\ell(v, w) \leq f(v, w) \leq u(v, w)$ for each $(v, w) \in \mathcal{E}$. It is known that if all capacities are integer and the set of feasible circulations is non-empty, then an integer-valued feasible circulation exists and can be found in polynomial time [17].

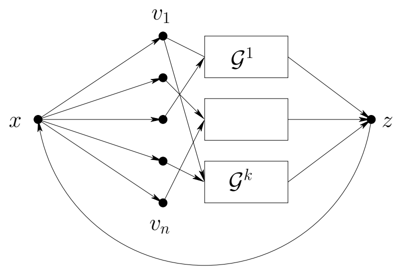

Figure 1: An instance of the circulation problem

All instances of the circulation problem that will be constructed in this paper have similar structure: They have two special nodes x and z (intuitively, the "source" and the "sink"), a node v_i for each voter $i \in N$ and a "gadget" (a node or a collection of interconnected nodes) $\mathcal{G}^j$ for each issue $j \in \mathcal{I}$. For each $i \in N$ there is an arc (x, v_i) with $\ell(x, v_i) = 0$ and $u(x, v_i) = 1$; a unit of flow on this arc is interpreted as an indication that voter i is assigned to *some* committee. The arc(s) from v_i to $\mathcal{G}^j$ encode voter i's preferences over A^j; the flow on such arcs indicates that voter i is assigned to the j-th committee. Each gadget $\mathcal{G}^j$ has arc(s) to z; the capacity constraints on these arcs are usually used to ensure that the j-th committee has exactly s^j members and votes in favor of $a^j_{i_j}$. Finally, there is a "backflow" arc from z to x with $\ell(z, x) = 0$, $u(z, x) = +\infty$. The overall structure of our circulation problem is illustrated in Figure 1. We will refer to the arcs of the form (x, v_i) and (z, x) as *generic*; in what follows, when describing an instance of the circulation problem, we only specify non-generic arcs and their capacities.

We are now ready to present our algorithm for binary domains.

THEOREM 3.2. *For every anonymous monotone poly-time computable voting rule $\mathcal{F}$ over $\{0, 1\}$ the problem $\mathcal{F}$-DELEGATION with $m = 2$ can be solved in time* $\text{poly}(n, k, s)$.

PROOF. Assume without loss of generality that the center's preferred outcome is $(1^1, \ldots, 1^k)$. Any anonymous monotone voting rule over $\{0, 1\}$ can be defined by a family of thresholds $\{T_n\}_{n\in\mathbb{N}}$: given an election with n voters, the rule outputs 1 if at least T_n voters prefer 1 to 0 and 0 otherwise [16, 6]; polynomial-time computability of $\mathcal{F}$ means that T_n can be computed efficiently given n. Consider an anonymous monotone voting rule $\mathcal{F}$ that corresponds to a family of thresholds $\{T_n\}_{n\in\mathbb{N}}$. Given an instance of $\mathcal{F}$-DELEGATION with $m = 2$, we construct an instance of the circulation problem as follows. The node set $\mathcal{V}$ of our directed graph is $\{x, z\} \cup \{v_i \mid i \in N\} \cup \mathcal{I}$. The arc set $\mathcal{E}$ contains all the generic arcs. Also, for each voter $i \in N$ and each issue $j \in \mathcal{I}$ such that i prefers 1^j to 0^j, $\mathcal{E}$ contains an arc (v_i, j) with $\ell(v_i, j) = 0$, $u(v_i, j) = 1$. Finally, for each $j \in \mathcal{I}$ the set $\mathcal{E}$ contains an arc (j, z) with $u(j, z) = +\infty$ and $\ell(j, z) = T_{s^j}$.

An integer feasible circulation in $(\mathcal{V}, \mathcal{E})$ corresponds to an assignment of "good" voters to committees: we have $(i, j) \in \mathcal{E}$ if and only if i's preferences on issue j coincide with those of the center. The lower bound on the arc (j, z) ensures that there sufficiently many "good" voters on the j-th committee. Each committee can then be filled up to capacity with the remaining voters in an arbitrary way. Conversely, it is not hard to see that any assignment of voters to the committees that satisfies the center can be converted into a feasible circulation. □

The restriction to binary domains is rather severe, and the reader may wonder if DELEGATION remains easy if m may be larger than 2, but is nevertheless bounded by a constant. We do not know if this is true in general; however, we can show that this is indeed the case if we additionally assume that k is bounded by a constant and $\mathcal{F}$ is anonymous and polynomial-time computable.

THEOREM 3.3. *For any anonymous polynomial-time computable voting rule $\mathcal{F}$ the problem $\mathcal{F}$-DELEGATION can be solved in time* $(s + 1)^{m!k} \cdot \text{poly}(n, m, k, s)$.

PROOF. The proof is similar to that of Theorem 3.2, but we will consider $(s + 1)^{m!k}$ instances of the circulation problem. All these instances will have the same underlying graph $\mathcal{G}$; however, the upper and lower bounds on the flow through some or the arcs will vary from one instance to another. The graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is constructed as follows. We set $\mathcal{V} = \{x, z\} \cup \{v_i \mid i \in N\} \cup \mathcal{W}$, where $\mathcal{W} = \{w_{j,t} \mid j \in \mathcal{I}, t = 1, \ldots, m!\}$. The nodes $w_{j,1}, \ldots, w_{j,m!}$ correspond to all possible orderings of the elements of A^j. For each $i \in N$ the node v_i has k outgoing arcs: there is an arc from v_i to $w_{j,t}$ with $\ell(v_i, w_{j,t}) = 0$, $u(v_i, w_{j,t}) = 1$ if the preferences of voter i over issue j are given by the ordering that corresponds to $w_{j,t}$. Also, $\mathcal{E}$ contains an arc from $w_{j,t}$ to z for each $w_{j,t} \in \mathcal{W}$, as well as all the generic arcs. To complete the description of the instance, it remains to describe the bounds on the flow along the arcs that lead from $\mathcal{W}$ to z.

We will say that a list of non-negative integers $\mathcal{K} = \{\kappa_{j,t} \mid j \in \mathcal{I}, t = 1, \ldots, m!\}$ is a *valid list of capacities* if

(1) $0 \le \kappa_{j,t} \le s$ for all $j \in \mathcal{I}$ and all $t = 1, \ldots, m!$,

(2) $\sum_{t=1,\ldots,m!} \kappa_{j,t} = s^j$ for each $j \in \mathcal{I}$, and

(3) for each $j \in \mathcal{I}$ it holds that in the election with the alternative set A^j where exactly $\kappa_{j,t}$ voters submit the preference ordering that corresponds to $w_{j,t}$ the $\mathcal{F}$-winner is $a^j_{i_j}$.

Condition (1) ensures that there are at most $(s + 1)^{m!k}$ valid lists of capacities; also, since $\mathcal{F}$ is polynomial-time computable and anonymous, condition (3) can be checked in time $\text{poly}(n, m, k, s)$. For each valid list of capacities $\mathcal{K}$, we complete the description of our circulation problem by setting $\ell(w_{j,t}, z) = u(w_{j,t}, z) = \kappa(j, t)$ for each $j \in \mathcal{I}$ and each $t = 1, \ldots, m!$. An integer feasible circulation in the resulting instance corresponds to an assignment of voters to committees in which each voter is assigned to at most one committee, the j-th committee has exactly s^j members, and the center is satisfied with the outcome of the vote in each committee. Thus, to solve $\mathcal{F}$-DELEGATION, it suffices to enumerate all valid lists of capacities and check if any of them corresponds to an instance of the circulation problem admitting a feasible solution. □

We remark that for some voting rules, such as, e.g., r-Approval with r bounded by a constant, we can modify the algorithm described in the proof of Theorem 3.3 so that the dependence of its running time on m becomes singly exponential (rather than doubly exponential, as in Theorem 3.3). Indeed, it is easy to see that for r-Approval, instead of creating a node for each of the possible $m!$ orderings of A^j, it suffices to create a node for each r-element subset of A^j, and the number of such subsets can be bounded by m^r. We obtain the following corollary.

COROLLARY 3.4. *r-Approval-DELEGATION can be solved in time* $(s + 1)^{m^r k} \cdot \text{poly}(n, m, k, s)$.

We will now briefly discuss what happens if we place restrictions on the number of committees k or the maximum committee size s.

Bounding k alone is unlikely to lead to efficient algorithms for our problem for general voting rules: in Section 6 we will show that $\mathcal{F}$-DELEGATION is NP-hard even for $k = 1$ for a number of common voting rules, including 3-Approval. However, for certain simple voting rules, such as Plurality and Veto, our problem can be solved efficiently if k is bounded by a constant (see Section 4). For 2-Approval, the complexity increases as we move from $k = 1$ to $k = 2$: in Section 5 we show that 2-Approval-DELEGATION is easy if $k = 1$, but becomes NP-hard if $k \geq 2$.

For small values of s, the situation is more complicated. If $s = 1$ (i.e., each committee consists of a single voter), $\mathcal{F}$-DELEGATION admits a very simple matching-based algorithm. Specifically, we construct a bipartite graph where the nodes on the left-hand side correspond to voters, the nodes on the right-hand side correspond to committees, and there is an edge from a voter i to a committee j if and only if $\mathcal{F}(R_i^j) = a^j_{i_j}$; clearly, a matching of size k in this graph corresponds to an assignment of voters to committees that makes the center happy. (Note that for committees of size 1 it is natural to assume that $\mathcal{F}$ outputs the unique voter's most preferred alternative, but this assumption is not necessary for our algorithm to work correctly). Thus, we obtain the following result.

PROPOSITION 3.5. *For $s = 1$ and any polynomial-time computable voting rule $\mathcal{F}$ the problem $\mathcal{F}$-DELEGATION can be solved in time* $\mathrm{poly}(n, m, k)$.

For some values of $s > 1$ we can still obtain easiness results for Plurality and Veto (see Section 4), but it is not clear if they can be extended to other voting rules. In fact, even assuming that both m and s are bounded by a constant does not seem to lead to efficient algorithms for our problem, even for the Plurality rule.

4. PLURALITY AND VETO

In this section, we present efficient algorithms for Plurality-DELEGATION and Veto-DELEGATION assuming that the number of issues k is bounded by a constant. We also prove easiness results for these two rules under some assumptions on the maximum committee size s (and, in the case of Plurality, on the tie-breaking rule).

THEOREM 4.1. *Both* Plurality-DELEGATION *and* Veto-DELEGATION *can be solved in time* $s^k \cdot \mathrm{poly}(n, m, k, s)$.

PROOF. The proof is similar to that of Theorems 3.2 and 3.3: we reduce our problem to solving at most s^k instances of the circulation problem. We will describe the construction for Plurality-DELEGATION; it is straightforward to modify our argument so that it applies to the Veto rule. For each of our instances, we set $\mathcal{V} = \{x, z\} \cup \{v_i \mid i \in N\} \cup \bigcup_{j\in\mathcal{I}} \left(A^j \cup \{z^j\}\right)$. In addition to the generic arcs, the arc set $\mathcal{E}$ contains an arc (z^j, z) with $\ell(z^j, z) = u(z^j, z) = s^j$ for each $j \in \mathcal{I}$. Also, for each $i \in N$ and each $j \in \mathcal{I}$ the set $\mathcal{E}$ contains an arc (v_i, a^j_t), where a^j_t is voter i's favorite alternative in A^j, with $\ell(v_i, a^j_t) = 0$, $u(v_i, a^j_t) = 1$. Finally, for each $j \in \mathcal{I}$ and each $a \in A^j$ there is an arc from a to z^j. It remains to set the bounds on the flow through these arcs; these will differ from one instance to another.

We will say that a list of non-negative integers $\mathcal{K} = \{\kappa^j \mid j \in \mathcal{I}\}$ is a *valid list of capacities* if $1 \leq \kappa^j \leq s^j$ for all $j \in \mathcal{I}$; note that there are at most s^k valid lists of capacities. Now, fix a valid list of capacities $\mathcal{K}$ and define the bounds on the flow through the arcs that lead into $z^1, \ldots, z^k$ as follows. For each $j \in \mathcal{I}$ and each $t = 1, \ldots, m$, set

- $\ell(a^j_t, z^j) = \kappa^j$, $u(a^j_t, z^j) = s^j$ if $t = i_j$;
- $\ell(a^j_t, z^j) = 0$, $u(a^j_t, z^j) = \kappa^j - 1$ if $t < i_j$, and
- $\ell(a^j_t, z^j) = 0$, $u(a^j_t, z^j) = \kappa^j$ if $t > i_j$.

A feasible circulation in the resulting network corresponds to an assignment of voters to committees in which (a) each voter is assigned to at most one committee, (b) the size of the j-th committee is exactly s^j, and (c) committee S^j contains at least κ^j voters that vote for $a^j_{i_j}$, and, for each $a \in A^j \setminus \{a^j_{i_j}\}$, the number of voters that vote for a is at most $\kappa^j - 1$ if the tie-breaking rule favors a over $a^j_{i_j}$ and at most κ^j otherwise. Clearly, under this assignment the center is satisfied. Thus, it suffices to go through all valid lists of capacities and check if any of them corresponds to an instance that admits a feasible circulation; the running time of this algorithm is $s^k \cdot \mathrm{poly}(n, m, k, s)$. □

An argument similar to the one given in the proof of Theorem 4.1 can be used to show that Plurality-DELEGATION is easy if $s \leq 3$ (while k can be arbitrary), under an additional assumption on the tie-breaking rules.

THEOREM 4.2. Plurality-DELEGATION *can be solved in time* $\mathrm{poly}(n, m, k)$ *if $s \leq 3$ and, furthermore, (a) for every $j \in \mathcal{I}$ such that $s^j = 3$ the ties are broken adversarially to the center (i.e., $i_j = m$), and (b) for every $j \in \mathcal{I}$ such that $s^j = 2$ the ties are broken either adversarially to the center or in the center's favor (i.e., $i_j = 1$ or $i_j = m$).*

PROOF. We will use the same graph as in the proof of Theorem 4.1, and argue that under the conditions of the theorem we only need to consider a single valid list of capacities. Indeed, if $s^j = 1$, we can ensure the desired outcome for issue j if and only if we have one vote in favor of $a^j_{i_j}$, so we set $\kappa^j = 1$. If $s^j = 2$ and $i_j = 1$, it suffices to ensure that $a^j_{i_j}$ gets at least one vote, as any tie will be broken in its favor, so we set $\kappa^j = 1$. If $s^j \in \{2, 3\}$ and $i_j = m$, $a^j_{i_j}$ needs exactly two votes to win, so we set $\kappa^j = 2$. It is not hard to check that an integer feasible circulation in the resulting network corresponds to an assignment of voters to committees that satisfies the center, and vice versa. □

Note that the argument on the proof of Theorem 4.2 does not go through if for some issue j with $s^j = 3$ the ties are broken in the center's favor. Indeed, in this case the center can get $a^j_{i_j}$ elected either by recruiting two voters who support $a^j_{i_j}$, or by recruiting one such voter and ensuring that the two other committee members support different alternatives; thus, we need to consider both $\kappa^j = 1$ and $\kappa^j = 2$. A similar difficulty arises if $s^j = 2$, but the tie-breaking rule is neither adversarial nor favorable to the center.

Observe also that Theorem 4.2 does not deal with Veto-DELEGATION. In fact, it is not clear if Veto-DELEGATION is easy for small values of s. However, interestingly, we can show that Veto-DELEGATION is in P as long as $s < m$; this holds even if s is *not* bounded by a constant.

THEOREM 4.3. *The problem* Veto-DELEGATION *can be solved in time* $\mathrm{poly}(n, m, k, s)$ *if* $s < m$.

PROOF. We say that a voter i on a committee S^j *vetoes* $a \in A^j$ if a is i's least preferred alternative in A^j. Fix an issue $j \in \mathcal{I}$ and a committee S^j. Since $s^j < m$, by the pigeonhole principle some alternative in A^j is *not vetoed* by any of the voters in S^j. Thus, $a^j_{i_j}$ is elected if and only if (a) no voter in S^j vetoes $a^j_{i_j}$ and (b) each alternative a^j_t with $t < i_j$ is vetoed by at least one voter in S^j. We will now use this observation in order to reduce Veto-DELEGATION to finding a feasible circulation. Our construction is similar to the one used in the proof of Theorem 4.1. Specifically, we build a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ as follows. We set $\mathcal{V} = \{x, z\} \cup$

$\{v_i \mid i \in N\} \cup \bigcup_{j\in\mathcal{I}} \left((A^j \setminus \{a^j_{i_j}\}) \cup \{z^j\} \right)$. In addition to the generic arcs, the arc set $\mathcal{E}$ contains an arc (z^j, z) with $\ell(z^j, z) = u(z^j, z) = s^j$ for each $j \in \mathcal{I}$. Also, for each $i \in N$ and each $j \in \mathcal{I}$ the set $\mathcal{E}$ contains an arc (v_i, a^j_t) if and only if a^j_t is voter i's least favorite alternative in A^j; we set $\ell(v_i, a^j_t) = 0$, $u(v_i, a^j_t) = 1$. Finally, for each $j \in \mathcal{I}$ and each $a^j_t \in A^j \setminus \{a^j_{i_j}\}$ there is an arc (a^j_t, z^j) such that $u(a^j_t, z^j) = s^j$ and, moreover, $\ell(a^j_t, z^j) = 1$ if $t < i_j$ and $\ell(a^j_t, z^j) = 0$ otherwise.

Clearly, an integer feasible circulation in this graph corresponds to an assignment of voters to committees where no voter is assigned to S^j if he vetoes $a^j_{i_j}$, the size of the j-th committee is s^j, and for each issue j it holds that every alternative that is favored over $a^j_{i_j}$ by the tie-breaking rule is vetoed by at least one voter. As argued above, under such an assignment the center's preferred outcome is achieved, and conversely, any assignment that satisfies the center can be converted into a feasible circulation in $\mathcal{G}$. $\square$

5. 2-APPROVAL

We will now consider a voting rule that is somewhat more complex than Plurality or Veto, namely, 2-Approval. For this rule, Lin [15] showed that the CCDV problem can be reduced to the problem of finding a b-matching (see [17] or the proof of Proposition 5.1 for a definition of b-matchings), and is therefore in P. This reduction can be modified to show that 2-Approval-DELEGATION with $k = 1$ is in P as well; for completeness, we provide a proof of this fact.

PROPOSITION 5.1. *For* $k = 1$, 2-Approval-DELEGATION *can be solved in time* $\mathrm{poly}(n, m, s)$.

PROOF. Let $A = A^1$ and suppose that the center's goal is to get a given alternative $a \in A$ elected by appointing a committee of size s. Let N' be the set of voters that rank a in the first two positions; we can assume without loss of generality that all voters in N' rank a first.

Suppose first that $|N'| \geq s$. Then if $s \geq 2$ and not all voters in N' rank the same alternative second, we are done: we identify two distinct alternative (say, b and c) that are ranked second by some voters in N', and pick s voters from N' so as to include at least one voter who ranks b second and at least one voter who ranks c second. In the resulting election a gets s 2-Approval points, while any other alternative gets at most $s - 1$ 2-Approval points, so a wins. If $s \geq 2$ and all voters in N' rank the same alternative (say, b) second, the outcome depends on how the ties are broken: if the tie-breaking rule favors a over b, the center can achieve its goal by appointing arbitrary s alternatives from N', and if the tie-breaking rule favors b over a, alternative a will not be elected no matter which committee is appointed. Finally, if $s = 1$, one has to check if there exists an alternative $b \in A \setminus \{a\}$ such that some voter in N' ranks b second and the tie-breaking rule favors a over b; if this is the case, the center can achieve its goal by appointing a voter who ranks a first and b second, and otherwise a cannot win.

Now, suppose that $|N'| < s$. In this case, our committee will include all voters in N' as well as some additional voters, which will be selected as follows. Recall that a *b-matching* in an undirected (multi)graph $G = (V, E)$ with given vertex capacities $\{u(v)\}_{v\in V}$ is a collection of edges $E' \subseteq E$ such that each vertex $v \in V$ is incident to at most $u(v)$ edges of E'. Given a bound B, we can decide in polynomial time whether a given (multi)graph with a given list of vertex capacities admits a b-matching of size at least B [17]. We will now show that our problem can be reduced to finding a b-matching of size $s - |N'|$ in a certain graph.

Specifically, consider a multigraph G that has $A \setminus \{a\}$ as its vertex set and contains an edge between b and c for each voter that ranks b and c in the top two positions (i.e., the number of parallel edges between b and c is the number of voters that rank b and c in the top two positions). For each $c \in A$, let r_c be the number of 2-Approval points that c receives from the voters in N'. For each vertex $c \in A \setminus \{a\}$ we set its capacity $u(c)$ to be $r_a - r_c$ if the tie-breaking rule favors a over c and $r_a - r_c - 1$ otherwise. It is not hard to see that a b-matching of size $s - |N'|$ in this multigraph corresponds to a set of voters that, together with the voters in N', form a committee that elects a. Conversely, if there exists a committee S, $|S| = s$, that gets a elected, our graph admits a b-matching of size $s - |N'|$; to see this, observe that we can assume without loss of generality that S contains all voters in N', and therefore the set of edges that correspond to voters in $S \setminus N'$ has to satisfy the capacity constraints. $\square$

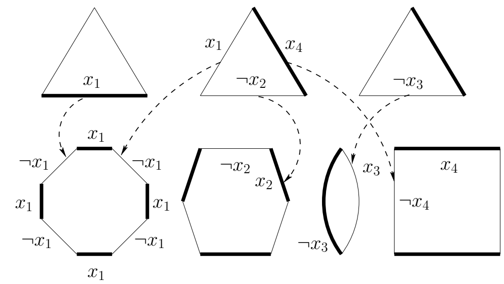

Figure 2: Fragments of the graphs G_1 (top) and G_2 (bottom) constructed in the proof of Theorem 5.2. Bold edges show a pair of matchings that correspond to the truth assignment $x_1 = \top, x_2 = \top, x_3 = \bot, x_4 = \top, \ldots$. The dashed arrows show the bijection μ.

However, if $k > 1$, 2-Approval-DELEGATION becomes NP-hard. To show this, we proceed in two steps: We first define a graph-theoretic problem, which we show to be NP-hard by a reduction from a variant of 3-SAT, and then we reduce this graph-theoretic problem to 2-Approval-DELEGATION.

An instance of our graph-theoretic problem, which we will call P-MATCHING, is given by two undirected graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ such that $|E_1| = |E_2|$, a bijection $\mu : E_1 \to E_2$, and two thresholds τ_1 and τ_2. It is a "yes"-instance if we can find two matchings $E'_1 \subseteq E_1$ and $E'_2 \subseteq E_2$ such that $|E'_1| \geq \tau_1$, $|E'_2| \geq \tau_2$ and $e \in E'_1$ implies $\mu(e) \notin E'_2$. P-MATCHING is a special case of the NP-hard MULTIPLE CHOICE MATCHING problem [11]; however, we were not able to obtain a direct reduction from MULTIPLE CHOICE MATCHING to P-MATCHING (or reduce MULTIPLE CHOICE MATCHING to 2-Approval-DELEGATION), so we will now give a different NP-hardness proof for our problem.

PROPOSITION 5.2. P-MATCHING *is* NP-*complete.*

PROOF. It is easy to see that this problem is in NP. To prove NP-hardness, we reduce from a variant of EXACT 3-SAT. Recall that an instance of EXACT 3-SAT is given by a set of Boolean variables $X = \{x_1, \ldots, x_q\}$ and a collection of clauses Cl, where each clause $cl \in Cl$ is a disjunction of exactly three literals (i.e., variables from X or their negations). It is a "yes"-instance if we can set the value of each variable in X to $\top$ ("true") or $\bot$ ("false") so that each clause in $\mathcal{C}$ is satisfied (i.e., contains at least one literal that is set to $\top$); otherwise it is a "no"-instance. EXACT 3-SAT is known to be NP-hard [11]. This holds even for the restricted version of this problem known as BALANCED EXACT 3-SAT (which we will abbreviate to BE 3-SAT), where we additionally require that for each $i = 1, \ldots, q$ the literals x_i and $\neg x_i$ occur the same number of times in Cl.

We will now reduce BE 3-SAT to P-MATCHING. Given an instance (X, Cl) of BE 3-SAT with $X = \{x_1, \dots, x_q\}$ and $Cl = \{cl_1, \dots, cl_r\}$, we construct an instance of P-MATCHING as follows. The graph G_1 consists of r cycles of length 3. For convenience, we label the edges of the j-th cycle with the literals that appear in cl_j (here we use the fact that each clause in Cl contains exactly three literals). The graph G_2 consists of q cycles $C_1, \dots, C_q$. The length of the i-th cycle is $2d_i$, where d_i is the number of occurrences of x_i in Cl. We number the edges of C_i clockwise (starting from an arbitrary edge), and label the odd-numbered and even-numbered edges with x_i and $\neg x_i$, respectively. Note that since (X, Cl) is balanced, G_1 and G_2 have the same number of edges. Further, there is a natural bijection μ between the edges of G_1 and those of G_2. Namely, μ maps the edge of G_1 that corresponds to the j-th occurrence of x_i (respectively, $\neg x_i$) in Cl to the $2j$-th (respectively, $(2j-1)$-st) edge of C_i. Note that an edge labeled with x_i is mapped to an edge labeled with $\neg x_i$, and vice versa (see Figure 2). Finally, we set $\tau_1 = r$, $\tau_2 = \sum_{i=1,\dots,q} d_i$. We leave it to the reader to see that the reduction is correct. $\square$

We are now ready to prove the main hardness result of this section.

THEOREM 5.3. 2-Approval-DELEGATION *is* NP-*complete even if* $k = 2$.

PROOF. Clearly, 2-Approval-DELEGATION is in NP. To prove NP-hardness, we give a reduction from P-MATCHING. Given an instance $(G_1, G_2, \mu, \tau_1, \tau_2)$ of P-MATCHING with $G_1 = (V_1, E_1)$, $G_2 = (V_2, E_2)$, $V_1 = \{a_1, \dots, a_\ell\}$, and $V_2 = \{b_1, \dots, b_t\}$, we construct an instance of 2-Approval-DELEGATION with $k = 2$ as follows. Assume without loss of generality that $\ell \le t$. We set $A^1 = V_1 \cup \{a_{\ell+1}, \dots, a_t\} \cup \{a, a'\}$, $A^2 = V_2 \cup \{b, b'\}$. We then set $n = |E_1| + 2 = |E_2| + 2$, and construct n voters as follows. Let $E_1 = \{e_1, \dots, e_{n-2}\}$. Let $i \in \{1, \dots n-2\}$, and suppose that $e_i = (a_x, a_y)$ and $\mu(e_i) = (b_z, b_w)$. Then the i-th voter's top two alternatives in A^1 and A^2 are a_x, a_y and b_z, b_w, respectively. The voters $n-1$ and n have identical preferences: their top two alternatives in A^1 and A^2 are a, a' and b, b', respectively. Set $s^1 = |\tau_1|+1$ and $s^2 = |\tau_2|+1$. By renaming the alternatives if necessary we can assume that the tie-breaking rule for A^1 (respectively, A^2) favors a (respectively, b) over all other alternatives. Finally, let the center's preferred outcome be (a, b). We leave it to the reader to see that the reduction is correct. $\square$

6. 3-APPROVAL AND OTHER VOTING RULES: HARDNESS RESULTS

In this section, we show that for many voting rules, the DELEGATION problem is NP-hard even for $k = 1$. We will prove this formally for r-Approval with $r \ge 3$ by modifying the proof for CCDV given in [15]. We then discuss the implications of the known hardness results for related problems (CCDV and winner determination in lot-based voting rules) for our setting.

THEOREM 6.1. *For every* $r \ge 3$ *the problem* r-Approval-DELEGATION *is* NP-*complete even if* $k = 1$.

PROOF. We first give the proof for $r = 3$; later, we will show how to modify it for other values of r.

It is easy to see that 3-Approval-DELEGATION is in NP. To prove hardness, we give a reduction from EXACT COVER BY 3-SETS (X3C) [11]. Recall that an instance of X3C is given by a ground set $X = \{x_1, \dots, x_{3q}\}$ and a collection of subsets $\mathcal{C} \subseteq 2^X$, where $|C| = 3$ for each $C \in \mathcal{C}$. It is a "yes"-instance if there exists a subcollection $\mathcal{C}' \subseteq \mathcal{C}$ such that $|\mathcal{C}'| = q$ and $\cup_{C \in \mathcal{C}'} C = X$ and a "no"-instance otherwise.

Given an instance $(X, \mathcal{C})$ of X3C, we construct an instance of 3-Approval-DELEGATION with $k = 1$ as follows. We set $A = A^1 = \{x_1, \dots, x_{3q}, z, y_1, y_2\}$ and let $s = s^1 = q + 1$. For each $C \in \mathcal{C}$ we create a voter v_C that ranks the alternatives in C in top three positions (in any order); there is also a voter v_z that ranks z, y_1, and y_2 in the top three positions. We let z be the center's preferred alternative. By renaming alternatives if necessary, we can assume that the tie-breaking rule favors z over all other alternatives.

Clearly, for the committee to elect z it has to include voter v_z as well as q other voters. Moreover, no alternative other than z should receive two or more 3-Approval points, which means that these voters correspond to an exact cover of X. Conversely, if $\mathcal{C}'$ is an exact cover of X, then $\{v_C \mid C \in \mathcal{C}'\} \cup \{v_z\}$ is a committee of size s that gets z elected.

For $r > 3$, we can modify this proof by adding $|\mathcal{C}| + 1$ blocks of dummy alternatives of size $r - 3$ each so that the i-th voter ranks the dummy alternatives in the i-th block in positions $4, \dots, r$; the rest of the proof goes through unchanged. $\square$

Let LOT-THEN-$\mathcal{F}$ denote the problem of deciding whether a given alternative has a positive chance of winning an election under a lot-based voting rule that uses the voting rule $\mathcal{F}$ at the second stage. Walsh and Xia prove that LOT-THEN-Borda is NP-hard (Theorem 6 in [18]). This implies that Borda-DELEGATION is NP-hard even for $k = 1$. The hardness of Bucklin-CCDV is proved in [8] (Theorem 3.15); one can check that this proof also applies to Bucklin-DELEGATION with $k = 1$. Similarly, the existing hardness proofs for Copeland-CCDV (Theorems 4.18 and 4.19 in [10]) and Maximin-CCDV (Theorem 5.6 in [9]) can be used to prove the hardness of our problem for $k = 1$. Briefly, to show this, we observe that the proofs in [8], [10] and [9] proceed by a reduction from X3C, with voters corresponding to sets, and either the deleted voters or the surviving voters encoding an exact cover. Thus, the number of voters to be deleted can be read off the instance description, which means that these proofs apply to our setting as well.

7. EXTENSIONS

We will now briefly discuss several possible extension of our basic model.

Assigning Voters to Committees In the model described in Section 2 each voter can be assigned to at most one committee, and every voter can serve on every committee. However, all of our algorithmic results generalize to the more realistic setting where for each voter i there is a bound b_i on the number of committees she can serve on (in which case the constraint on the total size of all committees becomes $s^1 + \dots + s^k \le b_1 + \dots + b_n$), and for each voter i there is a subset of issues $\mathcal{I}_i$ such that i can only be assigned to a committee S^j with $j \in \mathcal{I}_i$. Indeed, we can replace a voter that can serve on b_i committees with b_i identical voters that can serve on one committee each; clearly, the new instance admits a solution if and only if the original one does. Further, if some "voter, committee" pairs are not allowed, we can modify our circulation-based and matching-based algorithms by removing the respective edges from the graph; as for enumeration-based algorithms (i.e., Proposition 3.1), we can simply ignore the assignments that contain "illegal" pairs. Also, our algorithms can still be used when the center does not get to select the entire committee, but rather has to fill a few slots on an existing committee (assuming that it knows the preferences of the already appointed committee members). We omit the detailed description of the modified algorithms due to space constraints.

Approval Preferences So far, we have assumed that the center can only be satisfied with a single outcome. However, in practice the center's preferences may be more complex: for instance, the center may accept several combinations of alternatives, i.e., it may be equally happy with any outcome in a set $\mathcal{X} \subseteq \mathcal{A}$. In this case, we will say that the center has *approval* preferences. We will now discuss how to extend the results of the previous sections to this setting. We will focus on binary domains (i.e., $m = 2$), both because it is one of the cases where DELEGATION admits an efficient algorithm (Theorem 3.2) and because there is a natural formalism for representing the center's preferences under this assumption (see below); also, we assume that our voting rule is Majority, i.e., the rule that outputs the alternative(s) supported by at least half of the voters (combined with some tie-breaking rule).

Observe first that even for $m = 2$ we have $|\mathcal{A}| = 2^k$, and therefore the size of the set $\mathcal{X}$ may be exponential in the number of committees. Thus, if we are interested in an efficient algorithm for committee selection in this model, we need a succinct way of describing the center's preferences. It will be convenient to use the language of Boolean formulas for this purpose. Specifically, we identify each issue $j \in \mathcal{I}$ with a Boolean variable ξ^j, and say that an outcome $(a^1, \dots, a^k)$ *satisfies* a Boolean formula ϕ over $\{\xi^1, \dots, \xi^k\}$ if ϕ is satisfied by the truth assignment that for each $j \in \mathcal{I}$ sets $\xi^j = \top$ ("true") if $a^j = 1^j$ and $\xi^j = \bot$ ("false") if $a^j = 0^j$. Then every formula ϕ over $\{\xi^1, \dots, \xi^k\}$ naturally defines a set of outcomes $\mathcal{X}_\phi = \{\mathbf{a} \in \mathcal{A} \mid \mathbf{a} \text{ satisfies } \phi\}$. Clearly, for every set $\mathcal{X} \subseteq \mathcal{A}$ there exists a Boolean formula ϕ such that $\mathcal{X} = \mathcal{X}_\phi$, i.e., this language is complete for representing approval preferences over binary domains; while the size of ϕ is not guaranteed to be polynomial in k, there are many interesting classes of approval preferences that can be represented by small formulas. We remark that this model fits into the framework of prioritized goals for preference modelling (see, e.g., [14]).

An instance of the center's computational problem in this setting, which we will refer to as APP-DELEGATION, is described in the same way as an instance of DELEGATION (restricted to the binary domain), with one exception: instead of specifying the center's preferred outcome $(a^1, \dots, a^k)$, we specify a Boolean formula that encodes a set $\mathcal{X} \subseteq \mathcal{A}$ as described above.

Not surprisingly, if we allow arbitrary Boolean formulas in the description of the center's preferences, APP-DELEGATION is computationally hard; this holds even if each committee is a singleton.

THEOREM 7.1. *APP-DELEGATION is* NP*-complete. This hardness result holds even if $s = 1$.*

PROOF. To see that APP-DELEGATION is in NP, observe that we can guess an assignment of voters to committees, determine the election outcome in each committee, and then check if the corresponding truth assignment satisfies ϕ.

To prove that this problem is NP-hard, we give a reduction from SAT. Recall that an instance of SAT is given by a set of Boolean variables $X = \{x_1, \dots, x_q\}$ and a Boolean formula ψ over X; it is a "yes"-instance if and only if ψ is satisfiable. Given an instance (X, ψ) of SAT, we construct an instance of APP-DELEGATION as follows. We set $k = q$, $s^1 = \dots = s^k = 1$. Further, we set $n = 2k$ and construct n voters as follows. Voter $2j - 1$, $j = 1, \dots, k$, prefers 0^t to 1^t for all $t = 1, \dots, k$. Voter $2j$, $j = 1, \dots, k$, prefers 1^j to 0^j and prefers 0^t to 1^t for all $t \neq j$. Finally, ϕ is obtained from ψ by replacing each occurrence of x_j with ξ^j for $j = 1, \dots, k$.

Suppose first that ψ is satisfiable, and let $x'_1, \dots, x'_q$ be a satisfying assignment for ψ. We then assign voters to committees as follows: for each $j = 1, \dots, k$, if $x'_j = \top$, we set $S^j = \{2j\}$, and if $x'_j = \bot$, we set $S^j = \{2j - 1\}$. It is easy to see that the resulting outcome satisfies the center. Conversely, if there is an assignment of voters to committees that satisfies the center, then ϕ is satisfiable, and hence so is ψ. $\Box$

On the other hand, if k is bounded by a constant, APP-DELEGATION can be reduced to Majority-DELEGATION (which is is in P by Theorem 3.2). Indeed, we can obtain an explicit list of outcomes in $\mathcal{X}_\phi$ by enumerating all 2^k truth assignments for ϕ, and then use our algorithm for Majority-DELEGATION to check if any of these outcomes can be achieved. This argument extends to the case where ϕ is given in disjunctive normal form (DNF), and its DNF consists of at most $\mathrm{poly}(k)$ conjunctions.

However, there are interesting scenarios that that are not captured by this approach. Suppose, for instance, that k is even, and the center is satisfied if for every pair of issues $(2j, 2j-1)$ at least one of the corresponding committees votes for 1; that is, $\phi = (\xi^1 \vee \xi^2) \wedge \dots \wedge (\xi^{k-1} \vee \xi^k)$. While the formula ϕ itself is easy to satisfy, it admits $3^{k/2}$ satisfying assignments, so it is not clear how we can efficiently find an assignment of voters to committees that makes the center happy. More generally, the complexity of APP-DELEGATION under natural restrictions on the formula ϕ (such as, e.g., monotonicity) is an intriguing question for future research.

Ordered Preferences Another natural way to generalize our basic model is to assume that the center's preferences are given by a total order $\succ$ over the set $\mathcal{A}$ of all possible outcomes; in this case, we will say that the center's preferences are *ordered*. In this setting, a natural goal for the center is to implement the best *feasible* outcome, i.e., identify an outcome $\mathbf{a}$ such that $\mathbf{a}$ can be achieved for some list of committees $(S^1, \dots, S^k)$, whereas no outcome $\mathbf{a}' \succ \mathbf{a}$ can be achieved.

In this section, we study a special case of this problem where the center's preferences are *lexicographic*. That is, we assume that for each issue $j \in \mathcal{I}$ the center has a preference ordering $\succ^j$ over A^j, and it prefers $(a^1, \dots, a^k)$ to $(b^1, \dots, b^k)$ if and only if there exists a $j \in \{1, \dots, k\}$ such that $a^t = b^t$ for $t < j$ and $a^j \succ^j b^j$; we will refer to this problem as LEX-DELEGATION. We will now argue that LEX-DELEGATION reduces to DELEGATION.

THEOREM 7.2. *For any voting rule $\mathcal{F}$ the problem $\mathcal{F}$-LEX-DELEGATION can be reduced to solving at most $(m-1)k$ instances of $\mathcal{F}$-DELEGATION.*

PROOF. To simplify the presentation, we assume that $m = 2$ and the center prefers 1^j to 0^j for each $j \in \mathcal{I}$.

We use binary search, and construct an outcome $(a^1, \dots, a^k)$ as we proceed. We start by asking if there is a way of assigning voters to committees so that for issue 1 alternative 1^1 is selected. This amounts to solving an instance of $\mathcal{F}$-DELEGATION that is obtained from our input instance of $\mathcal{F}$-LEX-DELEGATION by removing issues $2, \dots, k$ and the voters' preferences over $A^2, \dots, A^k$ from the description of the instance, and setting the center's desired outcome for the only remaining issue to be 1^1. If the answer is "yes", we set $a^1 = 1^1$, and otherwise we set $a^1 = 0^1$.

At step j, $j = 2, \dots, k$, we ask if there is a way of assigning voters to committees so that for each $t = 1, \dots, j - 1$ committee S^t selects the alternative a^t, and committee S^j selects 1^1. Again, this amounts to solving $\mathcal{F}$-DELEGATION for a reduced instance of the original problem. We set $a^j = 1$ if the answer is "yes" and $a^j = 0$ if the answer is "no".

In the end we obtain a feasible outcome $\mathbf{a} = (a^1, \dots, a^k)$. Now, suppose that there is a feasible outcome $\mathbf{b} = (b^1, \dots, b^k)$ that the center prefers to $\mathbf{a}$. This means that there exists a $j \in \{1, \dots, k\}$ such that $a^t = b^t$ for $t < j$ and $b^j = 1^j$, $a^j = 0^j$. Since $\mathbf{b}$ is feasible, the call to $\mathcal{F}$-DELEGATION at step j should have returned "yes"; this is a contradiction with $a^j = 0^j$.

For $m > 2$, at step j we consider all alternatives in A^j in the order of center's preferences, starting with the center's most preferred one. For each alternative $a \in A^j$ except for the last one we use a call to $\mathcal{F}$-DELEGATION to determine whether the vector $(a^1, \dots, a^{j-1})$ constructed so far can be extended with a. Note that we do not need to call $\mathcal{F}$-DELEGATION once we reach the center's least preferred alternative in A^j: there is always *some* way to extend $(a^1, \dots, a^{j-1})$. $\square$

More generally, the center's preferences can be described by some language for representing preferences over combinatorial domains, such as CP-nets [4] or weighted goals [5]; exploring whether Theorem 7.2 can be extended to such languages is an interesting direction for future work.

8. CONCLUSIONS AND FUTURE WORK

We have put forward a formal model for the problem of strategic delegation of decisions to committees that consist of voters with known preferences, and investigated the computational complexity of this problem for a number of voting rules, under various assumptions on the problem parameters. Our algorithmic results are summarized in Table 1. Of course, not all constraints on the values of the parameters listed in Table 1 are equally realistic, and hence not all of these results are equally important: while the number of committees k is likely to be fairly small in many settings (and hence Theorem 4.1 is likely to be quite useful), and decisions are often made over binary issues (in which case Theorem 3.2 applies), the number of voters n can often be fairly large (and hence the value of Proposition 3.1(a) is mostly theoretical). However, we believe that all results listed in Table 1 are valuable, as they contribute to providing a broader picture of the complexity of our problem.

Assumption	Rules	Reference
$n < C$	poly-time	Prop. 3.1(a)
$s, k < C$	poly-time	Prop. 3.1(b)
$m = 2$	anonymous, monotone, poly-time	Thm. 3.2
$m, k < C$	anonymous, poly-time	Thm. 3.3
$s = 1$	poly-time	Prop. 3.5
$k < C$	Plurality, Veto	Thm. 4.1
$s \leq 3$	Plurality (adv./fav. tie-breaking)	Thm. 4.2
$s < m$	Veto	Thm. 4.3
$k = 1$	2-Approval	Prop. 5.1

Table 1: Algorithmic results (C is a constant, n is the number of voters, k is the number of issues, m is the number of alternatives for each issue, and s is the maximum committee size).

Perhaps the most pressing open question is the complexity of Plurality-DELEGATION when k can be arbitrary; while this problem appears to be computationally difficult, we did not manage to show that it is NP-hard. There are also other values of the problem parameters for which we have neither polynomial-time algorithms nor NP-hardness results, such as, e.g., the case where both m and s are bounded by a constant.

Also, for the richer models of the center's preferences, such as the ones described in Section 7, our analysis is far from complete, and we have explicitly suggested some questions about these models that, in our opinion, are worth investigating. Further afield, it would be interesting to explore the setting where not only the center, but also the voters themselves have complex preferences, i.e., a voter's ranking of alternatives for issue j may depend on the alternative selected for issue j'.

Finally, one can ask what happens if the committee has to be appointed by two or more parties who nominate committee members subject to given quotas, either sequentially or simultaneously, giving rise to a strategic game; indeed, many real-life committees are formed in this manner.

Acknowledgements This research was supported by National Research Foundation (Singapore) Grant RF2009-08 (Elkind), Singapore MOE Tier 2 Grant MOE2011-T2-1-090 (ARC 19/11) (Pasechnik), and ERC Advanced Grant RACE (Wooldridge). Part of this research was done when the first two authors were visiting Oxford.

9. REFERENCES

[1] K. Arrow, A. Sen, and K. Suzumura, editors. *Handbook of Social Choice and Welfare, Volume 1*. Elsevier, 2002.

[2] J. Bartholdi, C. Tovey, and M. Trick. How hard is it to control an election? *Mathematical and Computer Modeling*, 16(8/9):27–40, 1992.

[3] E. Bonzon, M.-C. Lagasquie-Schiex, J. Lang, and B. Zanuttini. Boolean games revisited. In *ECAI'06*, pages 265–269, 2006.

[4] 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. *JAIR*, 21:135–191, 2004.

[5] Y. Chevaleyre, U. Endriss, and J. Lang. Expressive power of weighted propositional formulas for cardinal preference modeling. In *KR'06*, pages 145–152, 2006.

[6] F. Dietrich and C. List. Judgment aggregation by quota rules: Majority voting generalized. *Journal of Theoretical Politics*, 19:391–424, 2007.

[7] P. E. Dunne, W. van der Hoek, S. Kraus, and M. Wooldridge. Cooperative boolean games. In *AAMAS'08*, pages 1015–1022, 2008.

[8] G. Erdélyi, M. Fellows, L. Piras, and J. Rothe. Control complexity in Bucklin and Fallback voting. Technical Report arXiv:1103.2230, arXiv.org, 2011.

[9] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. Multimode control attacks on elections. *JAIR*, 40:305–351, 2011.

[10] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Llull and Copeland voting computationally resist bribery and constructive control. *JAIR*, 35:275–341, 2009.

[11] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, 1979.

[12] P. Harrenstein, W. van der Hoek, J.-J. Meyer, and C. Witteveen. Boolean games. In *TARK'01*, pages 287–298, 2001.

[13] S. Kraus and M. Wooldridge. Delegating decisions in strategic settings. In *ECAI'12*, pages 468–473, 2012.

[14] J. Lang. Logical preference representation and combinatorial vote. *AMAI*, 42(1–3):37–71, 2004.

[15] A. Lin. The complexity of manipulating k-approval elections. In *ICAART'11*, pages 212–218, 2011.

[16] K. O. May. A set of independent necessary and sufficient conditions for simple majority decisions. *Econometrica*, 20(4):680–684, 1952.

[17] A. Schrijver. *Combinatorial Optimization—Polyhedra and Efficiency*. Springer, 2003.

[18] T. Walsh and L. Xia. Lot-based voting rules. In *AAMAS'12*, pages 603–610, 2012.