A Broader Picture of the Complexity of Strategic Behavior in Multi-Winner Elections

Reshef Meir, Ariel D. Procaccia, Jeffrey S. Rosenschein

School of Engineering and Computer Science
The Hebrew University of Jerusalem
Jerusalem, Israel
{reshef24, arielpro, jeff}@cs.huji.ac.il

ABSTRACT

Recent work by Procaccia, Rosenschein and Zohar [14] established some results regarding the complexity of manipulation and control in elections with multiple winners, such as elections of an assembly or committee; that work provided an initial understanding of the topic. In this paper, we paint a more complete picture of the topic, investigating four prominent multi-winner voting rules. First, we characterize the complexity of manipulation and control in these voting rules under various kinds of formalizations of the manipulator's goal. Second, we extend the results about complexity of control to various well-known types of control. This work enhances our comprehension of which multi-winner voting rules should be employed in various settings.

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*;
J.4 [**Computer Applications**]: Social and Behavioral Sciences—*Economics*

General Terms

Algorithms, Theory, Economics

Keywords

Computational complexity, Voting

1. INTRODUCTION

Game theory is chiefly concerned with the reaction of rational entities to incentives. These entities can, naturally, be people (who, in practice, may not act rationally), but, alternatively, can be computational agents, driven by pristine calculations of utility. The theory of social choice, in particular, has long struggled with the following problem: is it possible to prevent strategic behavior on the part of the participants in an election (the voters, or the authority conducting the election)?

Cite as: A Broader Picture of the Complexity of Strategic Behavior in Multi-Winner Elections, R. Meir, A. D. Procaccia and J. S. Rosenschein, *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, Padgham, Parkes, Müller and Parsons (eds.),May,12-16.,2008,Estoril,Portugal,pp.991-998.

In an election the voters are asked to report their preferences over candidates. A voter is said to *manipulate* the election when he reports false preferences, in an attempt to influence the outcome of the election. The election's result is determined by a voting rule, which designates a winning candidate given the voters' preferences. The Gibbard-Satterthwaite theorem [9] asserts that any voting rule which cannot be manipulated must be a dictatorship, i.e., there is one voter who dictates the outcome of the election.[1] A considerable body of work has been devoted to circumventing this theorem. Some of the approaches considered in economics are restrictions of the agents' preferences, or assuming that money is available, which leads to mechanism design solutions.

An equally sinister setting is the one where the authority controlling the election, referred to as the *chairman*, attempts to *control* the outcome of the election by tampering with the list of registered voters or the slate of candidates. For instance, the chairman might register additional voters who support his favorite candidate, or demolish the competition by disqualifying some of the candidates.

1.1 A Computational Approach

In a series of important papers, Bartholdi, Tovey and Trick [2, 3] have argued that *computational complexity* can be a barrier against strategic behavior in elections. Indeed, although manipulation and control may be possible in theory, in practice they can amount to solving $\mathcal{NP}$-hard problems, suggesting that the voters or the chairman might as well avoid cheating altogether.

Indeed, in the context of manipulation, the well-known Single Transferable Vote (STV) rule has long been known to be $\mathcal{NP}$-hard to manipulate [1]. More recent papers show that common voting rules can be enhanced in a way that makes them hard to manipulate [4, 7], or that prominent rules are hard to manipulate in alternative settings, such as manipulation by coalitions of voters (see, e.g., [6]).

The computational aspects of control have also received attention, for the same reasons mentioned above. Inspired by Bartholdi, Tovey and Trick [3], very recent papers have extended their results to various types of control and a variety of voting rules [11, 10, 8].

[1] This is a simplification, as the theorem requires some assumptions. Most importantly, there must be at least 3 candidates, and the domain of the agents' preferences is the domain of all possible linear orders over the candidates.

1.2 Multi-winner elections and our results

The purpose of multi-winner elections is to choose a committee or assembly (for instance, elections for parliament). A multi-winner voting rule maps the preferences of the voters to subsets of candidates, and effectively specifies the composition of the assembly. We note that this structured setting, where voters essentially have preferences over subsets of candidates, is related to recent work on combinatorial voting (see, e.g., Lang [12] and the references listed there).

Now, the properties that are considered especially desirable with respect to multi-winner voting rules are not necessarily the ones usually sought in single-winner voting rules, and therefore different rules are usually investigated. The four prominent rules we shall examine here are Single Non-Transferable Vote (SNTV), Bloc voting, Approval, and Cumulative voting (for more details, see Section 2).

Some results have recently been established by Procaccia, Rosenschein and Zohar [14] regarding the complexity of manipulation and control in multi-winner elections. These results can be summarized as follows:

In...	MANIPULATION	CONTROL (Add Voters)
SNTV	$\mathcal{P}$	$\mathcal{P}$
Bloc	$\mathcal{P}$	$\mathcal{NP}$-c
Approval	$\mathcal{P}$	$\mathcal{NP}$-c
Cumulative	$\mathcal{NP}$-c	$\mathcal{NP}$-c

All the results hold for a very general formulation of the computational problems. The manipulator/chairman assigns utilities to different candidates; can he manipulate/control the election in a way that the total (additive) utility of the set of winners is above a given threshold? Our first goal is to extend all these results to more restricted questions. Is it possible to include a favorite candidate in the set of winners? Perhaps it is possible to determine completely the set of winners? Or maybe it is possible to just include as many favorite candidates as possible among the winners?

In addition, Procaccia et al. have only looked at one specific type of control, control by adding voters. We extend the results to three other types of control (introduced in Bartholdi et al. [3]), namely control by removing voters, by adding candidates, and by removing candidates. Impatient readers can jump directly to a summary of our results, available below as Tables 1 and 2.

2. PRELIMINARIES

We now introduce some notation, as well as the four multi-winner voting rules that we shall investigate.

Let the set of voters be $V = \{v_1, v_2, \dots v_n\}$; let the set of candidates be $C = \{c_1, c_2, \dots c_m\}$. We denote the number of seats in the assembly—the number of candidates to be elected—by $k \in \mathbb{N}$.

Multi-winner voting rules differ from single-winner rules in the properties that they are expected to satisfy. A major concern in multi-winner elections is *proportional representation*: a faction that consists of a fraction X of the population should be represented by approximately a fraction X of the seats in the assembly. This property is not satisfied by (generalizations of) many of the rules usually considered with respect to single-winner elections.

So, here we examine four of the prevalent multi-winner voting rules. In all four, the candidates are awarded points by the voters, and the candidates with the most points win the election.

- *Single Non-Transferable Vote* (SNTV): each voter gives one point to a favorite candidate.
- *Bloc voting*: each voter gives one point to each of k candidates.
- *Approval voting*: each voter can approve or disapprove any candidate; an approved candidate is awarded one point, and there is no limit to the number of candidates a voter can approve.
- *Cumulative voting*: allows voters to express intensities of preferences, by asking them to distribute a fixed number of points among the candidates. We denote the fixed pool of points by L.

Scoring rules are a prominent family of voting rules. A voting rule in this family is defined by a real vector $\vec{\alpha} = \langle \alpha_1, \dots, \alpha_m \rangle$, where $\alpha_l \geq \alpha_{l+1}$ for $l = 1, \dots, m-1$. Each voter reports a ranking of the candidates, thus awarding α_1 points to the top-ranked candidate, α_2 points to the second candidate, and in general α_l points to the candidate ranked in place l. Note that SNTV is the scoring rule defined by the vector $\langle 1, 0, \dots, 0 \rangle$, and Bloc is the scoring rule defined by vector $\langle 1, \dots, 1, 0, \dots, 0 \rangle$, where the number of 1's is k.

3. MANIPULATION

Manipulation in voting is considered to be any scenario in which a voter reveals false preferences in order to improve the outcome of the election. This has various negative consequences; not only do voters spend valuable computational resources determining which lie to employ, but worse, the outcome may not be one that reflects the social good. Presumably, a voting rule which is hard-to-manipulate *a priori* precludes such undesirable behavior. A general definition of the manipulation problem in multi-winner elections was given in Procaccia et al. [14]:

DEFINITION 3.1. In the (k-winner) MANIPULATION problem, we are given a set C of candidates, a set V of voters and their ballots, the number of winners $k \in \mathbb{N}$, a utility function $u : C \to \mathbb{Z}$, and an integer $t \in \mathbb{Z}$. We are asked whether a single additional voter (the manipulator) can cast his vote in a way that in the resulting election, $\sum_{c \in W} u(c) \geq t$, where W is the set of winners of size k.

REMARK 3.2. The Manipulator's utility function is implicitly assumed to be additive. One can consider more elaborate utility functions, such as the ones investigated in the context of combinatorial auctions, but that is beyond the scope of this paper.

Procaccia et al. [14] have established that this problem is tractable in SNTV, Bloc, and Approval, but that it is $\mathcal{NP}$-complete in Cumulative voting. Although Cumulative voting has emerged as the winner in this complexity-theoretic competition, one might argue that the general formulation of the problem given above makes manipulation harder. Indeed, the manipulator might have the following, more specific, goals in mind.

1. The manipulator has a specific candidate whom he is interested in seeing among the winners.

2. The manipulator has a favorite subset of candidates, and he is interested in seeing *all of them* among the winners.

3. The manipulator has a favorite subset of candidates, and he is interested in seeing *as many as possible of them* among the winners.

Notice that the third setting is a special case of Definition 3.1—indeed, simply restrict u to be a boolean-valued function, i.e., $u : C \to \{0,1\}$. Furthermore, the second setting is a special case of the third since, if $D \subseteq C$ is the favorite subset, we can set

$$u(c) = 1 \Leftrightarrow c \in D$$

and $t = |D|$. The first is a special case of the second when $|D| = 1$.

Remark 3.3. In all manipulation and control problems, we assume tie-breaking is *adversarial* to the manipulator or chairman, i.e., ties are broken in favor of candidates with lower utility. This is a standard assumption, made in many of the papers on these topics [6, 14].

The next proposition gives a negative answer to the question of whether Manipulation in Cumulative voting is still hard in the abovementioned settings. Indeed, we put forward an algorithm that decides the problem under any boolean-valued utility function.

Proposition 3.4. Manipulation *in Cumulative voting with any boolean-valued utility function* $u : C \to \{0,1\}$ *is in* $\mathcal{P}$.

Proof. Let $s[c]$ be the score of candidate $c \in C$ before the manipulator has cast his vote, and $s^*[c]$ be c's score when the manipulator's vote is taken into account. Assume without loss of generality that $s[c_1] \geq s[c_2] \geq \ldots \geq s[c_m]$. Let $D = \{d_1, d_2, \ldots\}$ be the set of desirable candidates $d \in C$ with $u(d) = 1$, and again assume these are sorted by nonincreasing scores.

Informally, we are going to find a threshold *thresh* such that pushing t candidates above the threshold guarantees their victory. Then we will check whether it is possible to distribute L points such that at least t candidates pass this threshold, where L is the number of points available to each voter.

Formally, consider Algorithm 1 (w.l.o.g. $k \geq t$, otherwise manipulation is impossible). The algorithm clearly halts in polynomial time. It only remains to prove the correctness of the algorithm.

Lemma 3.5. *The above algorithm correctly decides* Manipulation *in Cumulative voting with any boolean-valued utility function.*

Proof. Denote by $\hat{W} = \{c_1, \ldots, c_k\}$ the k candidates with highest score (sorted) before the manipulator's vote, and by W the final set of k winners. The threshold candidate c_{j^*} partitions $\hat{W}$ into two disjoint subsets: $\hat{W}_u = \{c_1, \ldots, c_{j^*-1}\}, \hat{W}_d = \{c_{j^*}, \ldots, c_k\}$. By the maximality of j^*, it holds that:

$$|\hat{W}_u \cap D| + |\hat{W}_d| = |\hat{W}_u \cap D| + (k + 1 - j^*) = t. \quad (1)$$

Note that S is the exact number of votes required to push t desirable candidates above the threshold. Now, we must show that the manipulator can cast his vote in a way that the winner set W satisfies $|W \cap D| \geq t$ if, and only if, $L \geq S$.

Algorithm 1 Decides Manipulation in Cumulative voting with boolean valued utility

$j^* \leftarrow max\{j : \ |\{c_1, c_2, \ldots, c_{j-1}\} \cap D| + k + 1 - j \geq t$ and $c_j \notin D\}$ ▷ j^* exists, since the condition holds for the first candidate not in D
thresh $\leftarrow s[c_{j^*}]$
$S \leftarrow \sum_{j=1}^{t} \max\{0, thresh + 1 - s[d_j]\}$
if $S \leq L$ **then**
  return true
else
  return false
end if

Suppose first that $S \leq L$. Then it is clearly possible to push t desirable candidates above *thresh*. $\hat{W}_u \cap D$ were above the threshold already; it follows that $\hat{W}_d$ was replaced entirely by desirable candidates.

Let $W = \{w_1, \ldots, w_k\}$ be the set of *new* winners. In particular, we can write $W = \hat{W}_u \uplus \{w_{j^*}, \ldots, w_k\}$. $\hat{W}_u$ contains $|\hat{W}_u \cap D|$ desirable candidates, while $\{w_{j^*}, \ldots, w_k\}$ consists purely of desirable candidates. By Equation (1):

$$\begin{aligned} |W \cap D| &= |\hat{W}_u \cap D| + |\{w_{j^*}, \ldots, w_k\}| \\ &= |\hat{W}_u \cap D| + |\hat{W}_d| \\ &= t \end{aligned}$$

Conversely, suppose $S > L$. We must show that the manipulator cannot distribute L points in a way that t candidates from D are among the winners.

Clearly there is no possibility to push t desirable candidates above *thresh*. Consider some ballot cast by the manipulator, and assume w.l.o.g. that the manipulator distributed points only among the candidates in D. Denote the new set of winners by $W = W_u \uplus W_d$, where

$$W_u = \{c \in C : \ s^*[c] > thresh\}$$

$$W_d = \{c \in C : \ s^*[c] \leq thresh\}.$$

We claim that

$$|W_u \cap \overline{D}| = k - t, \quad (2)$$

where $\overline{D} = C \setminus D$. Indeed, by Equation (1)

$$|\hat{W}_u \cap D| = t - k - 1 + j^*.$$

Since no votes were awarded to candidates in $\overline{D}$,

$$\begin{aligned} |W_u \cap \overline{D}| = |\hat{W}_u \cap \overline{D}| &= |\hat{W}_u| - |\hat{W}_u \cap D| \\ &= (j^* - 1) - (t - k - 1 + j^*) \\ &= k - t \end{aligned}$$

Denote by F the set of candidates that were pushed above the threshold. Formally:

$$F = \{c \in D : s[c] \leq thresh \text{ and } s^*[c] > thresh\}$$

Thus:

$$W_u = \hat{W}_u \uplus F.$$

Let w^* be the new position of candidate c_{j^*} when the candidates are sorted by nonincreasing $s^*[c]$. It holds that

$$w^* = j^* + |F|.$$

We now claim that

$$|W_d \cap \overline{D}| \geq 1. \tag{3}$$

Recall that there are less than t desirable candidates above the threshold, thus:

$$\begin{aligned}
|W_u \cap D| < t \quad &\Rightarrow \\
|\hat{W}_u \cap D| + |F| = |W_u \cap D| < t = |\hat{W}_u \cap D| + k + 1 - j^* \quad &\Rightarrow \\
|F| < k + 1 - j^* \quad &\Rightarrow \\
w^* = j^* + |F| < j^* + k + 1 - j^* = k + 1 \quad &\Rightarrow \\
w^* \leq k \quad &\Rightarrow \\
c_{j^*} \in W_d \quad &\Rightarrow \\
|W_d \cap \overline{D}| \geq 1 &
\end{aligned}$$

By combining Equations (2) and (3), we finally obtain:

$$\begin{aligned}
|W \cap D| &= k - |W \cap \overline{D}| \\
&= k - (|W_u \cap \overline{D}| + |W_d \cap \overline{D}|) \\
&\leq k - (k - t + 1) \\
&= t - 1 \\
&< t
\end{aligned}$$

□

The proof of Proposition 3.4 is completed. □

Remark 3.6. The proof shows that the manipulation of Cumulative voting by a *coalition* of (even weighted) voters, as in Conitzer et al. [6], is tractable under a boolean-valued utility function. This follows by simply joining the (weighted) score pools of all the voters in the coalition.

SNTV and Bloc voting, which are both scoring rules, are known to be easy to manipulate under a general utility function [14]. The next proposition establishes that this is true for any scoring rule, under a boolean-valued utility function.

Proposition 3.7. *Let P be a scoring rule defined by the parameters $\vec{\alpha} = \langle \alpha_1, \ldots, \alpha_m \rangle$. Manipulation in P with any boolean-valued utility function $u : C \to \{0, 1\}$ is in $\mathcal{P}$.*

Proof. Let $\vec{\alpha} = \langle \alpha_1, \ldots, \alpha_m \rangle$ be the parameters of the scoring rule in question. Denote the score of each candidate $c \in C$, before the manipulator has cast his vote, by $s[c]$. Let $\mathcal{J}$ be the manipulator's preference profile, given by:

$$\mathcal{J} = c_{j_1} \succ c_{j_2} \succ \ldots \succ c_{j_m}$$

Suppose some candidate $c \in C$ was ranked in place l by the manipulator, $c = c_{j_l}$. Denote the final score of candidate c, according to the manipulator's profile $\mathcal{J}$, by:

$$s_{\mathcal{J}}[c] = s[c] + \alpha_l$$

Finally, denote the winner set that results from the manipulator's ballot $\mathcal{J}$ by $W_{\mathcal{J}}$.

Lemma 3.8. *Given $C' \subseteq C$, $|C| = k$, it is possible to determine in polynomial time if there exists $\mathcal{J}$ s.t. $C' = W_J$.*

Proof. Denote $C' = \{c'_1 \ldots c'_k\}$,

$$C'' = C \setminus C' = \{c''_1, \ldots c''_{m-k}\},$$

where both C', C'' are sorted by nondecreasing score $s[c]$. Let

$$\mathcal{J}^* = c'_1 \succ c'_2 \succ \ldots \succ c'_k \succ c''_1 \succ \ldots \succ c''_{m-k}$$

This preference profile ranks the players in C' first, while giving more points to candidates with lower initial score. Candidates from C'' are ranked next, and the same rule applies. The intuition is that we would like the candidates in C' to have a high-as-possible, more or less balanced, score. Likewise, we would like the candidates in C'' to have a low-as-possible balanced score. This strategy generalizes the algorithm of Bartholdi et al. [2].

We claim that there exists $\mathcal{J}$ s.t. $C' = W_J$ iff $C' = W_{\mathcal{J}^*}$. If $C' = W_{\mathcal{J}^*}$ then obviously there exists $\mathcal{J}$ s.t. $C' = W_J$. Conversely, suppose there exists some $\mathcal{J}^\#$ such that $C' = W_{\mathcal{J}^\#}$. Without loss of generality, this holds (by the adversarial tie breaking assumption)[2] iff

$$\forall c' \in C', c'' \in C'',\ s_{\mathcal{J}^\#}[c''] < s_{\mathcal{J}^\#}[c']. \tag{4}$$

We argue that it is possible to obtain $\mathcal{J}^*$ from $\mathcal{J}^\#$ by iteratively transposing pairs of candidates, without changing the winner set. Indeed, we distinguish between three cases:

1. $\exists j_1, j_2 \in \{1, 2, \ldots, k\}$ such that $s[c'_{j_1}] > s[c'_{j_2}]$, but in $\mathcal{J}^\#$ it holds that $c'_{j_1} \succ c'_{j_2}$. Now, transpose the rankings of c'_{j_1} and c'_{j_2} in $\mathcal{J}^\#$, i.e., consider the preference profile which is identical to $\mathcal{J}^\#$ except that the places of c'_{j_1} and c'_{j_2} are switched. Denote by W the new set of winners.

   The score of c'_{j_2} increased, so he is certainly still in W. Moreover, the new final (possibly lower) score of c'_{j_1} is:

   $$s[c'_{j_1}] + \alpha_{j_2} \geq s[c'_{j_2}] + \alpha_{j_2} = s_{\mathcal{J}^\#}[c'_{j_2}]$$

   By (4) we have that:

   $$\forall c'' \in C'',\ s_{\mathcal{J}^\#}[c''] < s_{\mathcal{J}^\#}[c'_{j_2}]$$

   Therefore, $c'_{j_1} \in W$ even after the transposition. We conclude that it still holds that $C' = W$.

2. $\exists j_1, j_2 \in \{1, 2, \ldots, m-k\}$ such that $s[c''_{j_1}] > s[c''_{j_2}]$, but in $\mathcal{J}^\#$ it holds that $c''_{j_1} \succ c''_{j_2}$. A similar argument holds in this case.

3. $\exists c' \in C', c'' \in C''$ such that in $\mathcal{J}^\#$ it holds that $c'' \succ c'$. Clearly the desirable candidate c' can only rank higher if we transpose the two candidates.

Using the three types of transpositions, we can replace a couple of candidates at each step until we obtain $\mathcal{J}^*$ from $\mathcal{J}^\#$. In each such step it remains true that $C' = W$, thus $C' = W_{\mathcal{J}^*}$. □

Lemma 3.9. *Given $C' \subseteq C$, $C' \leq k$, it is possible to determine in polynomial time if there exists $\mathcal{J}$ s.t. $C' \subseteq W_{\mathcal{J}}$.*

Proof. Let $C' \subseteq C$, $|C'| = k' < k$. We add to C' the $k - k'$ candidates from C'' with the highest score (according to $s[c]$), and denote this new set of size k by C^*. According to Lemma 3.8, we can determine efficiently if there exists $\mathcal{J}$ such that $C^* = W_J$.

We argue that it is enough to check C^*. Indeed, assume that there exists $\mathcal{J}$ such that $C' \subseteq W_{\mathcal{J}}$. Let $c \in C \setminus W_{\mathcal{J}}$ such that there exists $c' \in W_{\mathcal{J}}$ with $s[c'] < s[c]$. Now, if we

[2]Tie breaking works against candidates with utility 1 (which are the ones we ultimately care about), but in favor of candidates in C' with utility 0. However, for ease of exposition, we do not deal with such borderline cases here.

transpose, in the ranking $\mathcal{J}$, the candidates c and c', clearly c becomes a winner while c' becomes a loser. Therefore, it is possible to make C^* the set of winners. □

To complete the proof of the proposition, we denote by D the set of candidates whose utility is 1. The total utility is at least t iff there is a subset of D of size t that can be included in W. Let C' be the t candidates with the highest score $s[c]$ in D. By similar arguments as before, if this subset cannot win then no other subset of D of size t can. By Lemma 3.9 we can efficiently find out whether it is possible to include C' in W. □

4. CONTROL

In the control setting, we assume that the authority controlling the election (hereinafter, the *chairman*) has the power to tweak the election's electorate or slate of candidates in a way that might change the outcome. This is also a form of undesirable strategic behavior, but on the part of a behind-the-scenes player who is not supposed to take an active part in the election.

Procaccia et al. [14] have explored one type of control, namely control by adding voters. In this setting, the chairman might add voters who support his candidate, but the number of voters he can add without alerting attention to his actions is limited. The problem is formally defined as follows:

DEFINITION 4.1. In the problem of CONTROL BY ADDING VOTERS, we are given a set C of candidates, a set V of registered voters, a set V' of unregistered voters, the number of winners $k \in \mathbb{N}$, a utility function $u : C \to \mathbb{Z}$, and integers $r, t \in \mathbb{N}$. We are asked whether it is possible to register at most r voters from V' such that in the resulting election, $\sum_{c\in W} u(c) \geq t$, where W is the set of winners, $|W| = k$.

It is known that CONTROL BY ADDING VOTERS is tractable in SNTV, and $\mathcal{NP}$-complete in Bloc voting, Approval voting, and Cumulative voting. As in Section 3, we will be interested in seeing if these results still hold in the special cases mentioned above (that is, if the results are true when the chairman only wants to get a specific candidate elected, wants the set of winners to be exactly a given set, or wants a given subset of candidates to be included in the set of winners).

Even more importantly, we would like to extend our results to some of the different types of control, first considered in Bartholdi et al. [3].

DEFINITION 4.2. In the problem of CONTROL BY REMOVING VOTERS, we are given a set C of candidates, a set V of registered voters, the number of winners $k \in \mathbb{N}$, a utility function $u : C \to \mathbb{Z}$, and integers $r, t \in \mathbb{N}$. We are asked whether it is possible to remove at most r voters from V such that in the resulting election, $\sum_{c\in W} u(c) \geq t$, where W is the set of winners, $|W| = k$.

Another possible misuse of the chairman's authority is tampering with the slate of candidates. Removing candidates is obviously helpful, but even adding candidates can sometimes tip the scales in the direction of the chairman's favorite.

DEFINITION 4.3. In the problem of *Control by Adding Candidates*, we are given a set C of registered candidates, a set C' of unregistered candidates, a set V of voters, the number of winners k, a utility function $u : C \cup C' \to \mathbb{Z}$, and integers $r, t \in \mathbb{N}$. All voters have preferences over all candidates $C \cup C'$. We are asked whether it is possible to add at most r candidates C'' from C', such that in the resulting elections on $C \cup C''$, $\sum_{c\in W} u(c) \geq t$, where W is the set of winners, $|W| = k$.

DEFINITION 4.4. In the problem of *Control by Removing Candidates*, we are given a set C of candidates, a set V of voters, the number of winners k, a utility function $u : C \to \mathbb{Z}$, and integers $r, t \in \mathbb{N}$. We are asked whether it is possible to remove at most r candidates from C', such that in the resulting elections $\sum_{c\in W} u(c) \geq t$, where W is the set of winners, $|W| = k$.

Some clarification is in order. In the context of scoring rules, the assumption in the above two problems is that the voters have rankings of all the candidates in $C \cup C'$. Therefore, if some candidates are added or removed, the voters' preferences over the new set of candidates are still well-defined. The same goes for Approval: each voter approves or disapproves every candidate in $C \cup C'$. However, in the context of Cumulative voting, the problems of control by adding/removing candidates are *not* well-defined. Indeed, one would require a specification of how the voters distribute their points among every possible subset of candidates, and this would require a representation of exponential size.[3] Consequently, we do not consider control by adding or removing candidates in Cumulative voting.

4.1 Controlling the Set of Voters

As noted above, our agenda in this subsection is two-fold: determining whether the results of Procaccia et al. [14] survive the transition from general utility functions to boolean-valued utility, and extending the results to control by removing voters.

We start with SNTV. Procaccia et al. [14] show that CONTROL BY ADDING VOTERS in SNTV is in $\mathcal{P}$, even under a general utility function (and therefore, obviously, for any restricted boolean-valued function). We show that the same is true for CONTROL BY REMOVING VOTERS.

PROPOSITION 4.5. Control by Removing Voters *in SNTV under any utility function is in $\mathcal{P}$.*

PROOF. We will provide a polynomial-time reduction from CONTROL BY REMOVING VOTERS in SNTV to CONTROL BY ADDING VOTERS in SNTV. Since *Control by Adding Voters* in SNTV is in $\mathcal{P}$, this is sufficient to prove the proposition.

Given an instance (V, r, t, k, C, u) of CONTROL BY REMOVING VOTERS, define an equivalent instance

$$(U, U', r^*, t^*, k^*, C^*, u^*)$$

of CONTROL BY ADDING VOTERS. Set:

[3]It is possible to imagine compact representations, but that is beyond the scope of this paper.

$$\begin{aligned} C^* &= C \\ r^* &= r \\ t^* &= t - \sum_{c \in C} u(c) \\ k^* &= |C| - k \\ u^*(c) &= -u(c) \\ U' &= V \end{aligned}$$

For each voter v, let $f(v)$ be the candidate that voter v ranks first; $f(v)$ gives all the relevant information about voter v's ballot. The ballots of the voters in U are defined by the following rule. For each candidate $c \in C$,

$$|\{v \in U : f(v) = c\}| = |V| - |\{v \in V : f(v) = c\}|$$

We claim that (V, r, t, k, C, u) is in CONTROL BY REMOVING VOTERS iff $(U, U', r^*, t^*, k^*, C^*, u^*)$ is in CONTROL BY ADDING VOTERS.

Let U'' be the subset of voters selected by the chairman from $U' = V$. Denote by $s[c], s^*[c]$ the final score of candidate c in the election obtained by removing or adding voters, respectively. It holds that:

$$s[c] = |\{v \in V : f(v) = c\}| - |\{v \in U'' : f(v) = c\}|.$$

Furthermore,

$$\begin{aligned} s^*[c] &= |\{v \in U : f(v) = c\}| + |\{v \in U'' : f(v) = c\}| \\ &= |V| - |\{v \in V : f(v) = c\}| + |\{v \in U'' : f(v) = c\}| \\ &= |V| - (|\{v \in V : f(v) = c\}| - |\{v \in U'' : f(v) = c\}|) \\ &= |V| - s[c] \end{aligned}$$

Hence for all $c, c' \in C$:

$$s^*[c] \geq s^*[c'] \Leftrightarrow s[c] \leq s[c']$$

We conclude that the $k^* = |C| - k$ winners of the constructed instance are exactly the $|C| - k$ losers of the original instance.[4] That is, if W are the winners in the given instance and W^* the winners in the constructed instance, we have that $W^* = C \setminus W$.

Finally,

$$\begin{aligned} \sum_{c \in W} u(c) - t &= \sum_{c \in C \setminus W^*} u(c) - (t^* + \sum_{c \in C} u(c)) \\ &= - \sum_{c \in C \setminus W^*} u^*(c) + \sum_{c \in C} u^*(c) - t^* \\ &= \sum_{c \in W^*} u^*(c) - t^* \end{aligned}$$

Thus, for any choice of subset U'' to be removed or added, $\sum_{c \in W} u(c) \geq t$ if, and only if, $\sum_{c \in W^*} u^*(c) \geq t^*$. We conclude that the given instance is a "yes" instance iff the constructed instance is a "yes" instance. □

We now tackle the other three voting rules. The results in Procaccia et al. [14], although formulated for a general utility function, actually establish the following lemma.

[4]This conclusion also takes into account the adversarial tie-breaking assumption.

LEMMA 4.6. *[14] CONTROL BY ADDING VOTERS in Bloc voting, Approval voting, and Cumulative voting is $\mathcal{NP}$-hard, even when the chairman simply wants to exclude a distinguished candidate.*

It is possible to provide a generic reduction that establishes the hardness of CONTROL BY ADDING VOTERS in all three rules, even when the voters just want to include a specific candidate among the winners. This, as usual, shows hardness in all the special cases we discussed.

PROPOSITION 4.7. CONTROL BY ADDING/REMOVING VOTERS *in Bloc voting, Approval voting, and Cumulative voting is $\mathcal{NP}$-hard, even when the chairman simply wants to include a distinguished candidate.*

PROOF. Omitted due to space constraints. □

Notice that the result for Approval follows from Hemaspaandra et al. [10]: constructive control by adding voters is known to be hard even in single-winner elections. Interestingly, they also show that in single-winner elections *destructive* control is easy, in contrast with our multi-winner results.

4.2 Controlling the Set of Candidates

We now turn to the problem of controlling the set of candidates. As noted at the beginning of this section, this problem is ill-defined when it comes to Cumulative voting, so in the following we restrict ourselves to SNTV, Bloc voting, and Approval voting.

First, we recall that Bartholdi et al. [3] show that control by adding/removing candidates in SNTV is $\mathcal{NP}$-complete, even in single winner elections (in particular, even when the chairman wants to include a single candidate among the winners). This result extends to Bloc voting:

PROPOSITION 4.8. CONTROL BY ADDING/REMOVING CANDIDATES *in Bloc voting is $\mathcal{NP}$-complete, even in single-winner elections.*

PROOF. Simply consider the case of $k = 1$. □

Although Approval voting seems more complicated than SNTV, surprisingly it is much easier to control by tampering with the set of candidates.

PROPOSITION 4.9. CONTROL BY ADDING CANDIDATES *in Approval is in $\mathcal{P}$, under any utility function.*

PROOF. We will actually solve the following problem: can the chairman add *exactly* r candidates, in a way that all the added candidates become winners, and the utility is at least t? Solving this problem entails that the chairman can also solve the original problem, as he can simply run the algorithm for every $s \leq r$.

First note that each candidate in $c \in C, C'$ has a fixed number of points $s[c]$, regardless of the participation of any other candidate. The chairman selects a subset $D \subseteq C'$, $|D| = r$, and the winners are the k candidates in $C \uplus D$ with the highest score. Therefore, it is only effective to add candidates that will actually be winners.

Say indeed that $D \subseteq W$. Regardless of the identity of the candidates in D, the winners from C are exactly the $|C| - r$ candidates with the highest score in C. Since r is fixed and (without loss of generality) $r \leq k$, the total score of the winners from C is also fixed, and we only need to optimize D, i.e., select the best r candidates from C' that can be made to be winners.

LEMMA 4.10. *Let $C = \{c_1, \ldots, c_m\}$ be sorted by nonincreasing score, then by nondecreasing utility, $s[c_j] \geq s[c_{j+1}]$ for all $j = 1, \ldots, m-1$, and if $s[c_j] = s[c_{j+1}]$ then $u(c_j) \leq u(c_{j+1})$. Let W be the set of winners after the candidates $D \subseteq C'$ have been added. Then $D \subseteq W$ iff for all $d \in D$, one of the following holds:*

1. $s[d] > s[c_{k-r+1}]$.

2. $s[d] = s[c_{k-r+1}]$ *and* $u(d) < u(c_{k-r+1})$.

PROOF. Assume first that $D \subseteq W$. Among the k candidates with highest score, at least r are not from C. Thus, at least r candidates among the highest-scoring k candidates in C are excluded from W; the candidates $\{c_{k-r+1}, \ldots, c_k\}$ are certainly excluded from the set of winners. Since candidates with lower score are excluded first, and equality is broken in favor of candidates with lower utility, we have that either $\forall c \in W$, $s[c_{k-r+1}] < s[c]$, or an equality holds and the utility of c is lower; in particular, this is true for any $d \in D \subseteq W$.

Conversely, suppose that for all $d \in D$, either condition 1 or condition 2 holds. Then exactly k candidates are preferred (by the voting rule and the tie-breaking mechanism) to c_{k-r+1}: $\{c_1, \ldots, c_{k-r}\}$, as well as the candidates in D. Thus these are the k winners. □

Denote $\mathit{thresh}(r) = s[c_{k-r+1}]$. By Lemma 4.10, it is sufficient to consider the candidates

$$\{c \in C' : s[c] > \mathit{thresh}(r) \text{ or } s[c] = \mathit{thresh}(r) \wedge u(c) < u(c_{k-r+1})\}.$$

Clearly, it is possible to achieve a utility of t iff this is accomplished by adding the r candidates with highest utility in this set. □

PROPOSITION 4.11. CONTROL BY REMOVING CANDIDATES *in Approval is in $\mathcal{P}$, under any utility function.*

PROOF. We describe an immediate polynomial-time reduction from CONTROL BY REMOVING CANDIDATES in Approval to CONTROL BY ADDING CANDIDATES in Approval. We reduce the versions of the problems where exactly r candidates are to be removed/added. As noted in the proof of Proposition 4.9, this is sufficient.

Let (C, V, k, u, r, t) be an instance of CONTROL BY REMOVING CANDIDATES in Approval (with exact r). Set:

$$\begin{aligned} C^* &= \emptyset \\ C'^* &= C \\ r^* &= |C| - r \end{aligned}$$

We claim that $(C^*, C'^*, V, k, u, r^*, t)$ is an equivalent instance of CONTROL BY ADDING CANDIDATES in Approval. Indeed, adding r^* candidates from C to the empty set is clearly equivalent to removing $r = |C| - r^*$ candidates from C. □

5. CONCLUSIONS

The analysis in this paper has focused on the complexity of manipulating and controlling four simple voting schemes that are often considered in the context of multi-winner elections: SNTV, Bloc voting, Approval voting, and Cumulative voting. We have mainly concentrated on answering the important questions that were left open by Procaccia et al. [14].

It was previously known that the multi-winner manipulation problem is tractable in the first three rules, but hard in Cumulative voting. Here we examined the complexity of the problem when the manipulator has a boolean-valued function (which is a generalization of, for example, the setting where the manipulator wants one candidate to be included among the winners). We showed that in this setting, manipulation is tractable even in Cumulative voting. Moreover, we have established that when the utility function is boolean-valued, the manipulation of *any* scoring rule is easy.

At this point we wish to direct the reader's attention to Table 1, which summarizes our results regarding manipulation. A left (respectively right) implication arrow in the table means that the result in the cell is implied by the result in the left (respectively right) cell. Notice that in the case of boolean-valued utilities, manipulation is tractable in all four voting schemes analyzed in this paper. This is not the case, however, with Single Transferable Vote (STV).

The (single-winner) version of STV works as follows. The voters report rankings of the candidates. The election proceeds in rounds. In the first round, the candidate whom the least number of voters ranked first is eliminated. The voters who voted for the eliminated candidate transfer their votes to the second candidate in their ranking. The election proceeds this way for $m-1$ rounds, until only a single candidate prevails. This voting rule can easily be generalized to multi-winner elections, by holding only $m-k$ rounds. In practice, STV is used for multi-winner elections, but is more complicated (different counting methods and quotas are employed), and herein lies one of its main weaknesses. Nevertheless, STV's weakness is also its strength: due to this voting rule's complexity, it is also $\mathcal{NP}$-hard to manipulate, even in single winner elections [3]. To conclude this point, our results show that the four prevalent multi-winner voting rules are susceptible to manipulation, if the manipulator is bent on simply including a candidate or a favorite subset among the winners. In settings where this is a concern, a multi-winner version of STV may be considered appropriate.

Table 2 summarizes our results regarding control. Notice that with respect to control, all results are true for the three types of utility functions that appear in the table. Previous work indicated that control by adding voters is easy in SNTV but hard in the other three rules. In this paper, we have shown that these results extend to control by removing voters. Surprisingly, the situation has turned on its head when it comes to control by adding or removing candidates: here it is Approval voting that is easy to control, while other rules are hard. Therefore, while the results of Procaccia et al. [14] indicated a clear hierarchy of resistance to strategic behavior, our extended results shatter this conception. We conclude one has to adopt a voting rule *ad hoc*, depending on whether control by tampering with the set of voters, or with the set of candidates, is the major concern.

Finally, we wish to connect this work with the ongoing discussion of worst-case versus average-case complexity of manipulation and control in elections. A strand of recent

MANIPULATION

In...	Include candidate	Boolean utility	General utility
SNTV	$\mathcal{P}$ $\Leftarrow$	$\mathcal{P}$ $\Leftarrow$	$\mathcal{P}$ [14]
Bloc	$\mathcal{P}$ $\Leftarrow$	$\mathcal{P}$ $\Leftarrow$	$\mathcal{P}$ [14]
Approval	$\mathcal{P}$ $\Leftarrow$	$\mathcal{P}$ $\Leftarrow$	$\mathcal{P}$ [14]
Cumulative	$\mathcal{P}$ $\Leftarrow$	$\mathcal{P}$	$\mathcal{NP}$-c [14]
STV	$\mathcal{NP}$-c [1]	$\Rightarrow$ $\mathcal{NP}$-c	$\Rightarrow$ $\mathcal{NP}$-c

Table 1: The complexity of manipulation in multi-winner elections

CONTROL

In...	Add/Remove Voters			Add/Remove Candidates		
	Include candidate	Boolean utility	General utility	Include candidate	Boolean utility	General utility
SNTV	$\mathcal{P}$ [10] $\Leftarrow$	$\mathcal{P}$ $\Leftarrow$	$\mathcal{P}$	$\mathcal{NP}$-c [3]	$\Rightarrow$ $\mathcal{NP}$-c	$\Rightarrow$ $\mathcal{NP}$-c
Bloc	$\mathcal{NP}$-c	$\Rightarrow$ $\mathcal{NP}$-c	$\Rightarrow$ $\mathcal{NP}$-c	$\mathcal{NP}$-c	$\Rightarrow$ $\mathcal{NP}$-c	$\Rightarrow$ $\mathcal{NP}$-c
Approval	$\mathcal{NP}$-c [10]	$\Rightarrow$ $\mathcal{NP}$-c	$\Rightarrow$ $\mathcal{NP}$-c	$\mathcal{P}$ $\Leftarrow$	$\mathcal{P}$ $\Leftarrow$	$\mathcal{P}$
Cumulative	$\mathcal{NP}$-c	$\Rightarrow$ $\mathcal{NP}$-c	$\Rightarrow$ $\mathcal{NP}$-c	Irrelevant	Irrelevant	Irrelevant

Table 2: The complexity of control in multi-winner elections

research argues that worst-case hardness is not a strong enough guarantee of resistance to strategic behavior, by showing that manipulation problems that are known to be $\mathcal{NP}$-hard are tractable according to different average-case notions [5, 13, 15]. However, these works are still highly inconclusive. Therefore, worst-case complexity of manipulation and control remains an important benchmark for comparing different voting rules, and still inspires a considerable and steadily growing body of work [6, 11, 10, 8].

6. ACKNOWLEDGMENT

This work was partially supported by Israel Science Foundation grant #898/05. Ariel Procaccia is supported by the Adams Fellowship Program of the Israel Academy of Sciences and Humanities.

7. REFERENCES

[1] J. Bartholdi and J. Orlin. Single transferable vote resists strategic voting. *Social Choice and Welfare*, 8:341–354, 1991.

[2] J. Bartholdi, C. A. Tovey, and M. A. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6:227–241, 1989.

[3] 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.

[4] V. Conitzer and T. Sandholm. Universal voting protocol tweaks to make manipulation hard. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 781–788, 2003.

[5] V. Conitzer and T. Sandholm. Nonexistence of voting rules that are usually hard to manipulate. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence*, pages 627–634, 2006.

[6] V. Conitzer, T. Sandholm, and J. Lang. When are elections with few candidates hard to manipulate? *Journal of the ACM*, 54:1–33, 2007.

[7] E. Elkind and H. Lipmaa. Hybrid voting protocols and hardness of manipulation. In *16th Annual International Symposium on Algorithms and Computation*, Lecture Notes in Computer Science, pages 206–215. Springer-Verlag, 2005.

[8] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Llull and Copeland voting broadly resist bribery and control. In *The 22nd National Conference on Artificial Intelligence*, pages 724–730, 2007.

[9] A. Gibbard. Manipulation of voting schemes. *Econometrica*, 41:587–602, 1973.

[10] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Anyone but him: The complexity of precluding an alternative. *Artificial Intelligence*, 171(5–6):255–285, 2007.

[11] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Hybrid elections broaden complexity-theoretic resistance to control. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 1308–1314, 2007.

[12] J. Lang. Vote and aggregation in combinatorial domains with structured preferences. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence*, pages 1366–1371, 2007.

[13] A. D. Procaccia and J. S. Rosenschein. Junta distributions and the average-case complexity of manipulating elections. *Journal of Artificial Intelligence Research*, 28:157–181, February 2007.

[14] A. D. Procaccia, J. S. Rosenschein, and A. Zohar. Multi-winner elections: Complexity of manipulation, control and winner-determination. In *Proceedings of the Twentieth International Joint Conference on Artificial Intelligence*, pages 1476–1481, 2007.

[15] M. Zuckerman, A. D. Procaccia, and J. S. Rosenschein. Algorithms for the coalitional manipulation problem. In *The ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 277–286, 2008.