Complexity of Control by Partitioning Veto and Maximin Elections

Cynthia Maushagen
Institut für Informatik
HHU Düsseldorf
40225 Düsseldorf, Germany

Jörg Rothe
Institut für Informatik
HHU Düsseldorf
40225 Düsseldorf, Germany

Abstract
Control by partition refers to situations where an election chair seeks to influence the outcome of an election by partitioning either the candidates or the voters into two groups, thus creating two first-round subelections that determine who will take part in a final round. In particular, “gerrymandering” (maliciously resizing election districts) can be modeled by partition-of-voters control attacks. While the complexity of control by partition (and other control actions) has been studied thoroughly for many voting systems, such results about the important systems veto and maximin voting are sparse. We settle the complexity of control by partition for veto in a broad variety of models and for maximin with respect to destructive control by partition of candidates.

1 Introduction
Along with manipulation (Bartholdi, Tovey, and Trick 1989; Conitzer, Sandholm, and Lang 2007) and bribery (Faliszewski, Hemaspaandra, and Hemaspaandra 2009; Faliszewski et al. 2009), electoral control (Bartholdi, Tovey, and Trick 1992; Hemaspaandra, Hemaspaandra, and Rothe 2007) has been the focus of much attention in computational social choice; see the book chapters by Faliszewski and Rothe (2015) and Baumeister and Rothe (2015) for a survey of the related results. Control scenarios model settings where an external agent, commonly referred to as the chair, seeks to influence the outcome of an election by such actions as adding, deleting, or partitioning either the candidates or the voters. We here focus on control by partition.

The above-mentioned chapters and the papers cited therein comprehensively describe applications of voting in artificial intelligence, multiagent systems, ranking algorithms, meta-websearch, etc., and they discuss how computational complexity can be used to provide some protection against manipulation, bribery, and control attacks. In particular, they give real-world examples of the various control types introduced by Bartholdi, Tovey, and Trick (1992) for the constructive control goal where the chair aims at making a given candidate win and by Hemaspaandra, Hemaspaandra, and Rothe (2007) for destructive control where the goal is to prevent a given candidate’s victory.

The complexity of control has been studied for many voting systems, including plurality, Condorcet, and approval voting (Bartholdi, Tovey, and Trick 1992; Hemaspaandra, Hemaspaandra, and Rothe 2007) and its variants (Erdélyi, Nowak, and Rothe 2009; Erdélyi et al. 2015), Copeland (Faliszewski et al. 2009), Borda (Russel 2007; Elkind, Faliszewski, and Slinko 2011; Loreggia et al. 2015; Chen et al. 2015), (normalized) range voting (Menton 2013), and Schulze voting (Parkes and Xia 2012; Menton and Singh 2013). Perhaps a bit surprisingly, complexity results about controlling the important systems veto and maximin voting are sparse. They have been investigated only with respect to control by adding or deleting candidates or voters: Faliszewski, Hemaspaandra, and Hemaspaandra (2011) studied maximin and Lin (2012) studied veto for these control types in terms of their computational complexity, and their parameterized complexity has been explored by Liu and Zhu (2010) for maximin and by Chen et al. (2015) for veto. To the best of our knowledge, complexity results for control by partition have been missing for these two systems to date.

This is the more surprising as control by partition of voters can model gerrymandering (i.e., maliciously resizing election districts), a particularly natural control type known from the real world. One reason why these control scenarios have been neglected so far for veto and maximin may be that proofs for control by partition tend to be technical and challenging. We settle the complexity of control by partition for veto in a broad variety of models and for maximin with respect to destructive control by partition of candidates.

2 Preliminaries
In this section, we define the needed voting systems and control problems and give some background on computational complexity.

Elections, Veto, and Maximin Voting
An election is given by a pair \((C,V)\), where \(C\) is a set of candidates and \(V\) a list of the voters’ preferences over the candidates. We will consider only preferences that are linear orders (strict rankings) with the left-most candidate being the most preferred one. For example, a preference \(d\ c\ a\ b\) means that this voter prefers \(d\) to \(c\), \(c\) to \(a\), and \(a\) to \(b\).

We will consider two well-known voting systems: veto (a.k.a. antiplurality) and maximin (a.k.a. Simpson).

- In **veto**, every voter vetoes her least preferred candidate, which means that this candidate gets no point while all
other candidates receive one point from this voter, and whoever scores the most points wins. Veto is a prominent positional scoring protocol, a class of important voting systems that are based on the candidates’ positional scores; besides veto, this class contains, for example, the popular voting systems plurality and Borda count.

- By contrast, maximin voting is based on the pairwise comparisons between the candidates and belongs to the class of Condorcet-consistent voting rules. Given an election \((C, V)\), for any two candidates \(c, d \in C\), let \(N(c, d)\) denote the number of voters preferring \(c\) to \(d\). The maximin score of \(c\) is \(\min_{d \neq c} N(c, d)\), and whoever has the largest maximin score wins the election.

**Control Problems**

We consider control by partition of either candidates or voters, as defined by Bartholdi, Tovey, and Trick (1992) and—for destructive control—by Hemaspaandra, Hemaspaandra, and Rothe (2007). The definitions below have been used in many papers; we refer to the book chapters by Faliszewski and Rothe (2015) and Baumeister and Rothe (2015) for the formal definitions of all problems studied here and for real-world examples motivating each control scenario we are interested in. In each such control scenario, starting from a given election \((C, V)\) and a distinguished candidate \(c \in C\), we form two subelections—either \((C_1, V)\) and \((C_2, V)\) where \(C\) is partitioned into \(C_1\) and \(C_2\), or \((C, V_1)\) and \((C, V_2)\) where \(V\) is partitioned into \(V_1\) and \(V_2\)—whose winners move forward to a final round if they survive the given tie-handling rule: either ties-eliminate (TE) that requires that only unique winners of a first-round subelection move forward, or ties-promote (TP) that requires that all winners of a first-round subelection move forward.

Such a partition of either \(C\) or \(V\) is the chair’s control action, and the chair’s goal is either to ensure that the distinguished candidate \(c\) wins the final round (in the constructive case) or to prevent \(c\)’s victory (in the destructive case), where the final round is always held with all votes from \(V\).

In the case of candidate control, we further distinguish between run-off partition of candidates, where the winners of \((C_1, V)\) and \((C_2, V)\) surviving the tie-handling rule face each other in the final run-off, and partition of candidates, where the winners of \((C_1, V)\) surviving the tie-handling rule face all candidates of \(C_2\) in the final round.

For each such control scenario, we can define a decision problem. As an example, we formally define the decision problem associated with constructive control by partition of voters in model TE for some given voting system \(\mathcal{E}\):

1. A (weak) Condorcet winner is a candidate who defeats (ties or defeats) every other candidate in pairwise comparison. A Condorcet winner does not always exist, but when they do, they are unique, whereas there always exists a weak Condorcet winner, possibly more than one. A voting rule is Condorcet-consistent if it respects the Condorcet winner whenever one exists.

<table>
<thead>
<tr>
<th><strong>(\mathcal{E})-Constructive-Control-by-Partition-of-Voters-TE</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Given:</strong></td>
</tr>
<tr>
<td><strong>Question:</strong></td>
</tr>
</tbody>
</table>

The above problem (denoted by \(\mathcal{E}\)-CCPV-TE)—the shorthand of the other problems to be used later on will be clear from this example)—is defined in the unique-winner model. We will also consider the nonunique-winner model where the question is changed to ask whether \(c\) is a winner (possibly among several winners) of the final round, and we will always specify the winner model we are referring to.

For a control type \(\mathcal{C}\) (such as constructive control by partition of voters in model TE), an election system \(\mathcal{E}\) is said to be immune to \(\mathcal{C}\) if it is impossible for the chair to reach her control goal (e.g., to make the given candidate \(c\) a unique winner in the constructive case for the unique-winner model, or to ensure that \(c\) is not a winner in the destructive case for the nonunique-winner model) via exerting control of type \(\mathcal{E}\); otherwise, \(\mathcal{E}\) is said to be susceptible to \(\mathcal{C}\). It is easy to observe that the two voting systems we study here, veto and maximin, are susceptible to every type of control (in both winner models) we have defined above; due to space limitations we omit giving detailed examples verifying these claims. If an election system \(\mathcal{E}\) is susceptible to some control type \(\mathcal{C}\), it is common to study the computational complexity of the associated control problem: We say \(\mathcal{E}\) is vulnerable to \(\mathcal{C}\) if the control problem corresponding to \(\mathcal{C}\) can be solved in polynomial time, and we say \(\mathcal{E}\) is resistant to \(\mathcal{C}\) if \(\mathcal{C}\) is NP-hard.

**Computational Complexity**

We assume that the reader is familiar with the basic notions of computational complexity, such as the complexity classes \(P\) (deterministic polynomial time) and \(NP\) (nondeterministic polynomial time) and with the notions of \(NP\)-hardness and \(NP\)-completeness, based on the polynomial-time many-one reducibility. For more background, we refer to the book by Garey and Johnson (1979).

### 3 Controlling Veto Elections by Partition of Voters in Model TE

In this section, we show that it is easy to control veto elections by partition of voters in model TE. We start with the constructive case.

**Veto-CCPV-TE**

We show that veto is vulnerable to constructive control by partition of voters in model TE, in both winner models. Essentially, the polynomial-time algorithm used to prove Theorem 3.1 exploits the fact that, due to the TE model, control is impossible only if either there are two candidates and the distinguished candidate is not already a veto winner (in the
unique-winner model: is not already the only veto winner) of the given election, or there are more than two candidates and some candidate other than the distinguished candidate is not vetoed by any voter. In all other cases it is easy to find a successful partition that ensures the distinguished candidate’s victory.

**Theorem 3.1.** Veto-CCPV-TE is in P in both the unique-winner and the nonunique-winner model.

**Proof.** The following polynomial-time algorithm solves the problem. Given an election \((C, V)\) with \(n\) votes in \(V\) and a candidate \(c \in C\), it proceeds as follows:

1. If there are no more than two candidates, then if \(c\) already is a winner (in the unique-winner model: the only winner) of \((C, V)\), control is possible via the trivial partition \((V, \emptyset)\), so accept; otherwise, control is impossible, so reject.

2. Otherwise (i.e., if \(|C| > 2\)), if \(\text{score}(d) = n\) for some \(d \in C \setminus \{c\}\), control is impossible, so reject.

3. Otherwise (i.e., if \(|C| > 2\) and \(\text{score}(d) < n\) for all \(d \in C \setminus \{c\}\)), it is safe to accept, since control is possible via the partition \((V_1, V_2)\) of \(V\) that puts all voters who veto \(c\) into \(V_1\) and all other voters into \(V_2\).

The above algorithm obviously runs in polynomial time and is correct. This is obvious for step 1. Further, it is impossible for \(c\) to defeat the candidate \(d\) with \(\text{score}(d) = n\) in step 2 (as \(d\) scores the maximum number of points in each first-round subelection, no matter how \(V\) is partitioned, which makes it impossible for \(c\) to win alone in any subelection). And in step 3, no candidate from \(V_1\) can move to the final round, because either \(V_1\) is empty (in case no one vetoes \(c\)) or each of the at least two candidates other than \(c\) wins subelection \((C, V_1)\) with the same score and, therefore, will be eliminated in model TE. On the other hand, each candidate \(d \neq c\) is vetoed by at least one voter ending up in \(V_2\), whereas \(c\) is not vetoed by any voter in \(V_2\) and thus wins subelection \((C, V_2)\) and the final run-off. This argument applies to both the unique-winner and the nonunique-winner model. 

**Veto-DCPV-TE**

A similar algorithm works in the destructive case.

**Theorem 3.2.** Veto-DCPV-TE is in P in both the unique-winner and the nonunique-winner model.

**Proof.** Given an election \((C, V)\) and a distinguished candidate \(c\), our algorithm works as follows:

1. If \(|C| = 1\), control is impossible, so reject.

2. If \(|C| = 2\), determine the set of veto winners. If \(c\) wins alone, control is impossible, so reject. Otherwise, control is possible via the trivial partition \((V, \emptyset)\), so accept.

3. If \(|C| > 2\), it is safe to outright accept, since control is always possible: Fix some candidate \(d \neq c\) and partition \(V\) into \((V_1, V_2)\) such that \(V_1\) contains all voters vetoing \(d\) and \(V_2\) contains all remaining voters.

The above algorithm obviously runs in polynomial time and its correctness is straightforward for steps 1 and 2, while it follows for step 3 from the observation that if either \(c\) or \(d\) is vetoed by everyone then \((V_1, V_2)\) will be trivial (either \((\emptyset, V)\) or \((V, \emptyset)\)) and will thus prevent \(c\) from winning, and if neither \(c\) nor \(d\) is vetoed by everyone then there is a candidate \(e, c \neq e \neq d\), who ties for winner with \(c\) in \((C, V_1)\), while \(d\) ties-or-defeats \(c\) in \((V_2)\); in either case, \(c\) cannot move forward to the final round due to model TE. 

## 4 Control by Partition of Candidates in Veto Elections

We now turn to control by partition of candidates in veto elections, considering both constructive and destructive control, both tie-handling models, TE and TP, both the unique-winner and the nonunique-winner model, and the partition problems both with and without run-off.

**Veto-CCRPC-TE, Veto-CCPC-TE, Veto-CCPC-TP**

We start by showing that veto is resistant to constructive control by run-off partition of candidates in model TE, dealing with the unique-winner model in Theorem 4.1 and with the nonunique-winner model in Corollary 4.4.

**Theorem 4.1.** Veto-CCRPC-TE is NP-complete in the unique-winner model.

**Proof.** Membership of Veto-CCRPC-TE in NP is obvious. To show that it is NP-hard, we reduce from ONE-IN-THREE-3SAT*, an adaption from the well-known NP-complete problem ONE-IN-THREE-3SAT where the clauses of the given boolean formula do not contain any negated variables (Garey and Johnson 1979, p. 259):

**ONE-IN-THREE-3SAT**

**Given:** A set \(X\) of boolean variables, a set \(S\) of clauses over \(X\), each containing exactly three unnegated literals.

**Question:** Does there exist a truth assignment to the variables in \(X\) such that exactly one literal is set to true for each clause in \(S\)?

Let \((X, S)\) be an instance of ONE-IN-THREE-3SAT* with \(X = \{x_1, \ldots, x_m\}\) and \(S = \{s_1, \ldots, s_n\}\). Construct an election \((C, V)\) with distinguished candidate \(c \in C\) by defining \(C = X \cup \{c, w\}\), where elements of \(X\) from now on will also be viewed as candidates, and the list \(V\) of votes as follows:

<table>
<thead>
<tr>
<th># votes</th>
<th>preference for each</th>
</tr>
</thead>
<tbody>
<tr>
<td>(2n^2 + 1):</td>
<td>(w c \cdots x_i) (i \in {1, \ldots, m})</td>
</tr>
<tr>
<td>(n - 1):</td>
<td>(w \cdots c)</td>
</tr>
<tr>
<td>(1):</td>
<td>(c \cdots w S_j \setminus {x_i}) (j \in {1, \ldots, n}) and (x_i \in S_j)</td>
</tr>
<tr>
<td>(2n):</td>
<td>(w \cdots c S_j) (j \in {1, \ldots, n})</td>
</tr>
</tbody>
</table>

If a set of candidates occurs in such a vote, we tacitly assume a fixed ordering of its candidates in this preference. The dots in a vote represent all remaining candidates (in an arbitrary, fixed order). In particular, there are \(3n\) votes of the form \(c \cdots w S_j \setminus \{x_i\}\). If, say, clause \(S_1\) contains the literals \(x_2, x_3, x_7\), then the corresponding three votes are \(c \cdots w x_2 x_3, c \cdots w x_2 x_7, c \cdots w x_3 x_7\).
Candidate $w$ alone wins in election $(C, V)$, since the candidates score the following points:\footnote{Here and in the following, we omit a detailed argumentation of why certain candidates score a certain number of points in some election, due to space limitations and since these scores can be determined straightforwardly.}

$$\text{score}(c) = (2n^2 + 1)m + 3n + 2n^2,$$
$$\text{score}(w) = (2n^2 + 1)m + 3n - n - 1 + 2n^2,$$
$$\text{score}(x_i) \leq (2n^2 + 1)(m - 1) + n - 1 + 3n + 2n^2.$$

Obviously, the reduction can be computed in polynomial time. It remains to show that $(X, S)$ is a yes-instance of ONE-IN-THREE-3SAT if and only if $(C, V, c)$ is a yes-instance of Veto-CCRPC-TE.

$(\Rightarrow)$ If $(X, S)$ is a yes-instance of ONE-IN-THREE-3SAT, then there is a subset $U = \{u_1, \ldots, u_k\}$ of $X$ (renaming its elements for convenience) such that \[\left| U \cap S \right| = 1\] for each $j \in \{1, \ldots, n\}$. We claim that partitioning $C$ into $C_1 = U \cup \{c, w\}$ and $C_2 = C \setminus C_1$ ensures that $c$ is the only veto winner. To see this, note that the candidates in subelection $(C_1, V)$ have the following scores:

$$\text{score}(c) = (2n^2 + 1)m + 3n + 2n^2,$$
$$\text{score}(w) = (2n^2 + 1)m + n - 1 + 2n + 2n^2,$$
$$\text{score}(u_i) \leq (2n^2 + 1)(m - 1) + n - 1 + 3n + 2n^2.$$

For $c$ to win $(C_1, V)$ alone, we have to show that $\text{score}(c) > \text{score}(w)$ and $\text{score}(c) > \text{score}(u_i)$ for all $u_i \in U$:

1. First, $\text{score}(c) > \text{score}(w)$ is equivalent to $(2n^2 + 1)m + 3n + 2n^2 > (2n^2 + 1)m + n - 1 + 2n + 2n^2$, which in turn is equivalent to $3n > n - 1$; second, $\text{score}(c) > \text{score}(u_i)$ is equivalent to $(2n^2 + 1)m + 3n + 2n^2 > (2n^2 + 1)(m - 1) + n - 1 + 3n + 2n^2$, which in turn is equivalent to $2n^2 + 1 > n - 1$.

Being the only veto winner of subelection $(C_1, V)$, $c$ will move forward to the final run-off. If more than one candidate wins subelection $(C_2, V)$ (thus TE blocking them all from moving to the final run-off), $c$’s overall victory is ensured. On the other hand, if some candidate $x_i \in C_2$ is the only veto winner of $(C_2, V)$, $c$ will face $x_i$ in the run-off. However, since

$$\text{score}(c) \geq (2n^2 + 1)m + 3n - n - 1 + 2n^2 \geq \text{score}(x_i)$$

in the run-off $\{(c, x_i), V\}$, $c$ wins the run-off and is the only overall veto winner. Thus $(C, V, c)$ is a yes-instance of Veto-CCRPC-TE in the unique-winner model.

$(\Leftarrow)$ Conversely, let $(X, S)$ be a no-instance of ONE-IN-THREE-3SAT. Then, for each partition of $X$ into $X_1$ and $X_2$, let $k_i$ be the number of clauses containing $i$ literals from $X_i$. We have $1 \leq k_0 + k_2 + k_3 \leq n$, since we started from a no-instance of ONE-IN-THREE-3SAT. We will show that for each possible combination of the $k_i$ (corresponding to each possible partition of $X$), candidate $c$ cannot end up being the only veto winner. Note that a partition of $X$ induces a partition of $C = X \cup \{c, w\}$ into $C_1$ and $C_2 = C \setminus C_1$. It is enough to distinguish the three cases below, and in each case, we will show that $c$ is not the only veto winner.

\begin{itemize}
  \item **Case 1:** $C_1 = \{c, w\}$. Then $\text{score}(c) = 3n$ and $\text{score}(w) = (2n^2 + 1)m + n - 1 + 2n^2 \geq 4n^2 + n$, so $w$ is the only veto winner of this subelection, and since $c$ does not take part in the final run-off, $c$ will not be an overall winner.
  \item **Case 2:** $C_1$ contains $c$ and some elements of $X$ but not $w$. It is enough to show that $w$ is the only winner of the other subelection, $(C_2, V)$, since if $c$ wins $(C_1, V)$, then either $c$ is not promoted to the final round due to TE (if there are other winners) or $c$ loses the final round as we have seen in Case 1. In subelection $(C_2, V)$, for each $x_i \in C_2$, we have

$$\text{score}(w) \geq (2n^2 + 1)m + n - 1 + 2n^2 > (2n^2 + 1)(m - 1) + n - 1 + 3n + 2n^2 \geq \text{score}(x_i),$$

where the “greater than” follows from $2n^2 + 1 > 3n$, which is true for all $n > 1$. (For $n = 1$, however, we would have started from a yes-instance of ONE-IN-THREE-3SAT, which contradicts our assumption.) Thus $w$ is the only veto winner of $(C_2, V)$, which precludes $c$’s overall victory in this case.
  \item **Case 3:** $C_1$ contains $c$, $w$, and some elements of $X$. Distinguish the following three subcases.
    \begin{itemize}
      \item **Case 3.1:** $k_0 \geq 2$. In this case, we have

$$\text{score}(c) \leq (2n^2 + 1)m + 3n + (n - k_0)2n$$
$$\text{score}(w) \geq (2n^2 + 1)m + n - 1 + 2n^2.$$ \[\text{score}(c) \leq \text{score}(w)\] holds since (for $k_0 \geq 2$) the inequality $2n^2 + 1 \leq 2k_0n$ implies $(2n^2 + 1)m + 3n + (n - k_0)2n \leq (2n^2 + 1)m + n - 1 + 2n^2$.

**Case 3.2:** $k_0 = 1$. In this case, we have

$$\text{score}(c) \leq (2n^2 + 1)m + 3n + (n - k_0)2n$$
$$\text{score}(w) \geq (2n^2 + 1)m + n - 1 + 2(n - 1) + 2n^2.$$ \[\text{score}(c) \leq \text{score}(w)\] holds since $n = 0$ can again be excluded implies $\text{score}(c) \leq \text{score}(w)$ also in this case.

**Case 3.3:** $k_0 = 0$. Since we have a no-instance, at least one clause must contain at least two literals from $X_1$, so

$$\text{score}(c) = (2n^2 + 1)m + 3n + 2n^2$$
$$\text{score}(w) \geq (2n^2 + 1)m + n - 1 + 2n + 1 + 2n^2.$$ \[\text{score}(c) \leq \text{score}(w)\] follows since $3n + 2n^2 \leq 2n^2 + 3n$ implies $(2n^2 + 1)m + 3n + 2n^2 \leq (2n^2 + 1)m + n - 1 + 2n + 1 + 2n^2$.

By model TE, $c$ cannot move forward to the final round and thus cannot win the overall election. As we have shown that $c$ is not the only veto winner in any partition of the candidates, $(C, V, c)$ is a no-instance of Veto-CCRPC-TE. \qed
A minor tweak in the construction of the previous proof (namely, by having \( n \) instead of \( n - 1 \) votes of the form \( w \cdots c \), all else being equal) works for showing NP-hardness of Veto-CCPC-TE and Veto-CCPC-TP in the nonunique-winner model. The proofs are omitted due to space.

**Theorem 4.2.** Veto-CCPC-TE and Veto-CCPC-TP are NP-complete in the nonunique-winner model.

**Veto-DCRPC-TE and Veto-DCPC-TE**

Now we turn to the destructive variant of the previous problem, but now in both winner models. We again show resistance via a reduction from ONE-IN-THREE-3SAT*.

**Theorem 4.3.** Veto-DCRPC-TE is NP-complete in both the unique-winner and the nonunique-winner model.

**Proof.** Membership of both problems in NP is again obvious. For showing NP-hardness, we start with the unique-winner model. Let \((X, S)\) be an instance of ONE-IN-THREE-3SAT* with \(X = \{x_1, \ldots, x_m\}\) and \(S = \{S_1, \ldots, S_n\}\). Construct an election \((C, V)\) with \(C = X \cup \{c, v\}\), \(c \in C\) being the distinguished candidate, and the following list of votes:

<table>
<thead>
<tr>
<th># votes</th>
<th>preference for each</th>
</tr>
</thead>
<tbody>
<tr>
<td>3n + 1</td>
<td>( e ) w \cdots ( x_i ) ( j \in {1, \ldots, m} )</td>
</tr>
<tr>
<td>2n + 2</td>
<td>( c ) w \cdots ( w S_j ) ( j \in {1, \ldots, n} )</td>
</tr>
<tr>
<td>( n )</td>
<td>( c ) w \cdots w</td>
</tr>
<tr>
<td>1</td>
<td>( w \cdots c S_j \setminus {x_i} ), ( j \in {1, \ldots, n} ) and ( x_i \in S_j )</td>
</tr>
</tbody>
</table>

The reduction can be computed in polynomial time. It is easy to see that \( c \) is the only veto winner of election \((C, V)\):

\[
\begin{align*}
score(c) &= (3n + 1)m + (2n + 2)n + n + 3n, \\
score(w) &= (3n + 1)m + (2n + 2)n + 3n, \\
score(x_i) &\leq (3n + 1)(m - 1) + (2n + 2)n + n + 3n.
\end{align*}
\]

We claim that \((X, S)\) is a yes-instance of ONE-IN-THREE-3SAT* if and only if \((C, V, c)\) is a yes-instance of Veto-DCRPC-TE.

\((\Rightarrow)\) If \((X, S)\) is a yes-instance of ONE-IN-THREE-3SAT*, then there is a subset \(U\) of \(X\) such that \(|U \cap S_j| = 1\) for each \( j \in \{1, \ldots, n\} \). Partitioning \(C\) into \(C_1 = U \cup \{c, w\}\) and \(C_2 = C \setminus C_1\) ensures that \( c \) is not the only veto winner, since \(c\) and \(w\) have the same score in subelection \((C_1, V)\):

\[
\begin{align*}
score(c) &= (3n + 1)m + (2n + 2)n + n + 2n \\
score(w) &= (3n + 1)m + (2n + 2)n + 3n,
\end{align*}
\]

so, by model TE, \( c \) cannot move forward to the final round.

\((\Leftarrow)\) Conversely, let \((X, S)\) be a no-instance of ONE-IN-THREE-3SAT*. As in the proof of Theorem 4.1, we consider all possible partitions of \( C \) into \( C_1 \) and \( C_2 \) and show that \( c \) always is the only veto winner overall.

**Case 1:** \( C_1 = \{c, w\} \). Then \( \text{score}(c) = (3n + 1)m + (2n + 2)n + n \) and \( \text{score}(w) = 3n \), so \( c \) moves forward to the final round. If the other subelection, \((C_2, V)\), has more than one winner, TE blocks them all, so \( c \) wins. If \((C_2, V)\) has a unique winner, say \( x_i \), we have \( \text{score}(c) = (3n + 1)m + (2n + 2)n + n \) and \( \text{score}(x_i) \leq 3n \) in the final round, \((\{c, x_i\}, V)\), so \( c \) wins.

**Case 2:** \( C_1 \) contains \( c \) and some elements but not \( w \).

\[
\begin{align*}
\text{score}(c) &= (3n + 1)m + (2n + 2)n + n + 3n \\
\text{score}(x_i) &\leq (3n + 1)(m - 1) + (2n + 2)n + n + 3n
\end{align*}
\]

then imply that \( c \) scores at least \( 3n + 1 \) points more than any other and moves forward to the final round. If \((C_2, V)\) has more than one winner, \( c \) outright wins; if either \( w \) or some \( x_i \) wins in \((C_2, V)\), \( c \) wins the run-off as shown in Case 1.

**Case 3:** \( C_1 \) contains \( c \), \( w \), and some elements of \( X \). Rename the elements of \( U = C_1 \cap X \) by \( U = \{u_1, \ldots, u_t\} \). Let \( k \) be the number of clauses \( S_j \) such that \(|S_j \cap U| = 0\).

**Case 3.1:** \( k > 0 \). Then the scores in \((C_1, V)\) are:

\[
\begin{align*}
\text{score}(c) &\geq (3n + 1)m + (2n + 2)n + n + (2n - k), \\
\text{score}(w) &= (3n + 1)m + (2n + 2)(n - k) + 3n, \\
\text{score}(u_i) &\leq (3n + 1)(m - 1) + (2n + 2)n + n + 3n.
\end{align*}
\]

For \( c \) to win subelection \((C_1, V)\) alone, we need to show that \( \text{score}(c) > \text{score}(w) \) and \( \text{score}(c) > \text{score}(u_i) \) for each \( u_i \in U \). Simplifying the scores of \( c \) and \( w \), we get \( 2n^2 + 5n - 2k > 2n^2 + 5n - 2nk - 2k \), which is equivalent to \( 2nk > 0 \), which is true because \( k > 0 \) and \( n > 0 \). Obviously, \( c \) also wins out over each \( u_i \in U \), since simplifying their scores yields \( 2n + 1 > 2k \), which is true. In the run-off, \( c \) is either alone or faces some \( x_i \) (if \( x_i \) is the only veto winner of subelection \((C_2, V)\)). By the argument just given, \( c \) triumphs over \( x_i \) and is the only overall veto winner.

**Case 3.2:** \( k = 0 \). Since \((X, S)\) is a no-instance, there is at least one clause \( S_j \) with \(|S_j \cap U| \leq 2\) in this case. This implies the following scores in \((C_1, V)\):

\[
\begin{align*}
\text{score}(c) &\geq (3n + 1)m + (2n + 2)n + n + 2n + 1, \\
\text{score}(w) &= (3n + 1)m + (2n + 2)n + 3n, \\
\text{score}(u_i) &\leq (3n + 1)(m - 1) + (2n + 2)n + n + 3n.
\end{align*}
\]

Thus \( c \) is the only veto winner of subelection \((C_1, V)\) and (by the above arguments) wins also the final run-off alone. Hence, \((C, V, c)\) is a no-instance of Veto-DCRPC-TE. \( \square \)

It is known that for voting systems that always have at least one winner (such as veto), any type of destructive control in the unique-winner model polynomial-time disjunctively truth-table reduces to the same type of constructive control in the nonunique-winner model (Hemaspaandra, Hemaspaandra, and Rothe 2007, Footnote 5 on p. 257). Therefore, Theorem 4.3 implies the following.

**Corollary 4.4.** Veto-CCRCP-TE in the nonunique-winner model cannot be in P, unless \( P = NP \).

In both winner models, the problems DCRPC-TE and DCPC-TE are known to be identical for all voting systems (Hemaspaandra, Hemaspaandra, and Menton 2013, Thm. 8 on p. 386); the proofs can be found in the related technical report by Hemaspaandra, Hemaspaandra, and Menton (2012). Thus we immediately have from Theorem 4.3:

**Corollary 4.5.** Veto-DCRPC-TE is NP-complete in both the unique-winner and the nonunique-winner model.

**Veto-DCRPC-TP and Veto-DCPC-TP**

We next turn to the ties-promote model, TP. By slightly modifying the proof of Theorem 4.3, we will show resistance in both cases for the nonunique-winner model.
Theorem 4.6. Veto-DCRPC-TP and Veto-DCPC-TP are NP-complete in the nonunique-winner model.

Proof. Starting with Veto-DCRPC-TP, we only describe the differences with the construction given in the proof of Theorem 4.3. The only required change is that the votes of the form \( c \cdots w \) (see the third row) occur \( n - 1 \) instead of \( n \) times. The arguments showing the correctness of the construction then need to be adapted to model TP; the details are omitted here due to space limitations. Regarding Veto-DCPC-TP, note that DCRPC-TP and DCPC-TP are known to be identical problems in the nonunique-winner model for all voting systems (Hemaspaandra, Hemaspaandra, and Menton 2013, Thm. 8 on p. 386).

5 Destructive Control by Partition of Candidates in Maximin Elections

Finally, we turn to destructive control by partition of candidates in maximin elections, focusing on the unique-winner model. We start with the ties-eliminate model.

Maximin-DCRPC-TE and Maximin-DCPC-TE

While veto is vulnerable to both constructive and destructive control by partition of voters but not to the types of candidate control we have studied, maximin voting turns out to be vulnerable to destructive control by partition of candidates.

Theorem 5.1. In the unique-winner model, maximin-DCRPC-TE is in P.

Proof. Given an election \((C, V)\) with distinguished candidate \(c \in C\) as input, our polynomial-time algorithm for maximin-DCRPC-TE simply works as follows: If \(c\) is the Condorcet winner of \((C, V)\), control is impossible, so reject; otherwise, accept.

To see that the algorithm is correct, note that control is always possible if \(c\) is not a Condorcet winner of \((C, V)\): This means that there is at least one candidate, say \(d \in C\), such that \(N(d, c) \geq N(c, d)\). Now, partitioning \(C\) into \(C_1 = \{d\}\) and \(C_2 = C \setminus C_1\) ensures that \(d\) moves forward to the final run-off, and even if \(c\) emerges as the only maximin winner of the other subelection, \((C_2, V)\), and faces \(d\) in the run-off, \(c\) will not be the only maximin winner of the overall election. On the other hand, if \(c\) is the Condorcet winner of \((C, V)\), no partition of \(C\) can prevent \(c\) from being the only maximin winner of the overall election.

Again, we can apply the known result that DCRPC-TE equals DCPC-TE for all voting systems (Hemaspaandra, Hemaspaandra, and Menton 2013, Thm. 8 on p. 386).

Corollary 5.2. In the unique-winner model, maximin-DCPC-TE is in P.

Maximin-DCRPC-TP and Maximin-DCPC-TP

In the ties-promote model, TP, the algorithm used to prove Theorem 5.1 works as well, though the proof of correctness needs to be slightly adjusted. Note that, unlike in TE, DCRPC-TP and DCPC-TP are not known to coincide in the unique-winner model, though DCRPC-TP equals DCPC-TP in the nonunique-winner model (Hemaspaandra, Hemaspaandra, and Menton 2013, Thm. 8 on p. 386), as noted in the proof of Theorem 4.6.

Theorem 5.3. In the unique-winner model, both maximin-DCRPC-TE and maximin-DCPC-TE are in P.

Proof. Given an election \((C, V)\) with distinguished candidate \(c \in C\) as input, the simple polynomial-time algorithm for maximin-DCRPC-TE from the proof of Theorem 5.1 also works here: If \(c\) is the Condorcet winner of \((C, V)\), reject; otherwise, accept.

The proof of correctness is adjusted as follows. If \(c\) is the Condorcet winner of \((C, V)\), our destructive goal can again never be reached: No partition of \(C\) can prevent \(c\) from being the only maximin winner of the overall election. On the other hand, if \(c\) is not a Condorcet winner of \((C, V)\), we distinguish two cases: First, if \(c\) is a weak Condorcet winner of \((C, V)\), there exists a candidate, say \(d\), such that \(N(d, c) = N(c, d)\); partitioning \(C\) into \(C_1 = \{d\}\) and \(C_2 = C \setminus C_1\) ensures that \(c\) will not be the only maximin winner of the overall election. Second, if \(c\) is not even a weak Condorcet winner of \((C, V)\), there exists a candidate, say \(d\), such that \(N(d, c) > N(c, d)\); partitioning \(C\) into \(C_1 = \{c, d\}\) and \(C_2 = C \setminus C_1\) will ensure that \(d\) does not even win sub-election \((C_1, V)\). Obviously, this argument works both with and without run-off, i.e., both for maximin-DCRPC-TP and maximin-DCPC-TP.

6 Conclusions and Open Questions

We have studied the complexity of control by partition of voters or candidates for veto and destructive control by partition of candidates for maximin. For future work, we propose to also settle the complexity of constructive control by partition of candidates and of all cases of control by partition of voters for maximin. Regarding veto, the control complexity is still open for partition of voters in model TP and for a number of cases for partition of candidates. In particular, note that we have studied maximin only in the unique-winner model and that also for veto some issues involving the choice of the winner model remain open.

On a higher level, a quite challenging interesting open question is to completely characterize the class of scoring protocols in terms of control complexity (i.e., to establish dichotomy results for the various control types), as has been done by Hemaspaandra and Hemaspaandra (2007) for constructive coalitional weighted manipulation, by Betzler and Dorn (2010) and Baumeister and Rothe (2012) for the possible winner problem (a generalization of coalitional unweighted manipulation due to Konczak and Lang (2005)), and by Hemaspaandra, Hemaspaandra, and Schnoor (2014) for constructive control by adding voters. Finally, it would also be interesting to study veto and maximin with respect to the refined models of control by partition introduced by Erdélyi, Hemaspaandra, and Hemaspaandra (2015).

Acknowledgments

This work has been supported in part by DFG grant RO-1202/15-1.
References