Toward Completing the Picture of Control in Schulze and Ranked Pairs Elections

Cynthia Maushagen, David Niclaus, Paul Nüsken, Jörg Rothe and Tessa Seeger
Heinrich-Heine-Universität Düsseldorf, MNF, Institut für Informatik, Düsseldorf, Germany
{cynthia.maushagen, david.niclaus, paul.nuesken, rothe, tessa.seeger}@hhu.de

Abstract
Both Schulze and ranked pairs are voting rules that satisfy many natural, desirable axioms. Many standard types of electoral control (with a chair seeking to change the outcome of an election by interfering with the election structure) have already been studied. However, for control by replacing candidates or voters and for (exact) multimode control that combines multiple standard attacks, many questions remain open. We solve a number of these open cases for Schulze and ranked pairs. In addition, we fix a flaw in the reduction of Minton and Singh showing that Schulze is resistant to constructive control by deleting candidates and re-establish a vulnerability result for destructive control by deleting candidates. In some of our proofs, we study variants of s-t vertex cuts in graphs that are related to our control problems.

1 Introduction
Elections play a fundamental role in decision-making processes of societies. Both the Schulze method [Schulze, 2011; Schulze, 2023] and the ranked pairs method [Tideman, 1987] satisfy many natural, desirable axioms.

The Schulze method is a relatively new voting rule and has gained unusual popularity over the past decade due to its outstanding axiomatic properties. In the real world, organizations like the Wikimedia Foundation, Kubernetes, or the Debian Vote Engine [Schulze, 2023] have used it in their decision-making processes. Although winner determination with the Schulze method is fairly complicated compared to most other voting rules, it can still be done in polynomial time [Schulze, 2011; Sornat et al., 2021]. The ranked pairs method was specifically designed to satisfy the independence of clones criterion [Tideman, 1987]. In general, its axiomatic properties are as outstanding as Schulze’s. That ranked pairs is barely widespread might be due to the fact that winner determination strongly depends on the handling of ties: When using “parallel universe tie-breaking” [Conitzer et al., 2009], the winner determination problem is NP-complete. However, it becomes tractable when defining ranked pairs as a resolute rule by using a fixed tie-breaking method [Brill and Fischer, 2012].

We study electoral control where a so-called election chair (or, simply chair) attempts to change the outcome of an election by changing its structure. Common examples are adding, deleting, partitioning [Bartholdi III et al., 1992; Hemaspaandra et al., 2007], or replacing [Loreggia et al., 2015] candidates or voters. In addition to these control types, we also study multimode control [Faliszewski et al., 2011] where the chair can combine several attacks into one. For each corresponding electoral control problem, there is a constructive case [Bartholdi III et al., 1992] where the goal is to make a favored candidate win the election, and a destructive case [Hemaspaandra et al., 2007] where the chair’s aim is to prevent a despised candidate from winning. It is natural to assume that it is beneficial for a voting rule to be immune to control, i.e., it is impossible to change the outcome of an election by that control type. However, immunity to control types does not occur often. In fact, most voting rule are susceptible to control [Faliszewski and Rothe, 2016], i.e., control is possible in at least some instances. Computational intractability can then be seen as a form of resistance to control: If it is computationally hard for an agent to decide if the goal of the attack can be achieved, it may deter the attacker from spending resources on this task. On the other hand, not all forms of control are malicious. In many cases, deciding if a control action can be achieved in polynomial time is beneficial for deciding whether to allocate resources to, e.g., a voter drive or spawning new nodes in the context of large clusters. We call a voting rule vulnerable to a control type if the corresponding decision problem can be decided in polynomial time.

While many standard control types have already been studied for Schulze and ranked pairs (Table 1), control by replacing candidates or voters and exact multimode control remained open. There are many real world situations where a chair must adhere to a specific number of candidates, e.g., if the election’s size is predetermined and a fixed number of candidates are already nominated, forcing the chair to nominate exactly the missing number of candidates. The restriction to these cases is motivated by the impossibility to reward strategic behaviour with success.

The supplementary material for this paper [Maushagen et al., 2024] can be found here: https://arxiv.org/abs/2405.08956.
tion makes control impossible in some situations, where control by adding fewer candidates would be successful. Other (practical) settings include autonomous agents where, e.g., the size of the cluster is predetermined. A chair may be able to influence some part of the cluster, but is forced to adhere to the overall determined size (examples can be found in the supplementary material).

Related Work: Bartholdi et al. [1992] introduced constructive control types and Hemaspaandra et al. [2007] the corresponding destructive cases. Control by replacing candidates or voters was introduced by Loreggi et al. [2015], while Faliszewski et al. [2011] introduced and studied multimode control. Erdélyi et al. [2021] provide an extensive study and overview of various control problems, including replacing candidates or voters and also multimode control.2

Other types of strategic influence on elections are manipulation, where a voter or a group of voters state their preferences strategically (i.e., untruthfully), and bribery, where a controlling agent bribes voters to change their preferences (see, e.g., Bartholdi III and Orlin, 1991; Bartholdi III et al., 1989; Conitzer and Sandholm, 2006; Faliszewski et al., 2009a; Faliszewski et al., 2009b). Control, bribery, and manipulation have been studied for a wide range of voting rules, as surveyed by Faliszewski and Rothe [2016] (bribery and control) and Conitzer and Walsh [2016] (manipulation), see also [Bauernfeind and Rothe, 2015].

For Schulze and ranked pairs, Parkes and Xia [2012], Xia et al. [2009], Menton and Singh [2013], and Gaspers et al. [2013] studied constructive and destructive control by adding or deleting voters or candidates, bribery, and manipulation. Table 1 gives an overview of known results. Hemaspaandra et al. [2013] showed fixed-parameter tractability for bribing, controlling, and manipulating Schulze and ranked pairs elections with respect to the number of candidates and provided algorithms with uniform polynomial running time that are independent of the number of candidates. Menton and Singh [2013] also provided results on control by partition and runoff partition of candidates and partition of voters for Schulze and further showed some results for all Condorcet-consistent voting rules.

2 Preliminaries

An election is a pair \((C, V)\), where \(C = \{c_1, \ldots, c_m\}\) is a set of candidates and \(V = \{v_1, \ldots, v_n\}\) is a list of voters in form of preferences over all candidates in \(C\). In this paper, preferences are expressed as a strict linear order over \(C\), where voters rank the candidates in descending order from most to least preferred. We write \(c \succ_{v_i} d\) to express that a voter \(v_i \in V\) prefers candidate \(c\) over \(d\). When it is clear from the context, we omit \(!_{v_i}\), and simply write \(cd\). For a set of candidates

<table>
<thead>
<tr>
<th>Schulze</th>
<th>Constructive</th>
<th>Destructive</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC</td>
<td>NP-c.</td>
<td>?</td>
</tr>
<tr>
<td>DC</td>
<td>NP-c.</td>
<td>P</td>
</tr>
<tr>
<td>AV</td>
<td>NP-c.</td>
<td>NP-c.</td>
</tr>
<tr>
<td>DV</td>
<td>NP-c.</td>
<td>NP-c.</td>
</tr>
<tr>
<td>B</td>
<td>NP-c.</td>
<td>NP-c.</td>
</tr>
<tr>
<td>M</td>
<td>P</td>
<td>P</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Ranked pairs</th>
<th>Constructive</th>
<th>Destructive</th>
</tr>
</thead>
<tbody>
<tr>
<td>AC</td>
<td>NP-c.</td>
<td>NP-c.</td>
</tr>
<tr>
<td>DC</td>
<td>NP-c.</td>
<td>NP-c.</td>
</tr>
<tr>
<td>AV</td>
<td>NP-c.</td>
<td>NP-c.</td>
</tr>
<tr>
<td>DV</td>
<td>NP-c.</td>
<td>NP-c.</td>
</tr>
<tr>
<td>B</td>
<td>NP-c.</td>
<td>NP-c.</td>
</tr>
<tr>
<td>M</td>
<td>NP-c.</td>
<td>NP-c.</td>
</tr>
</tbody>
</table>

Table 1: Overview of complexity results for standard control (AC, DC, AV, DV), bribery (B), and manipulation (M) in Schulze and ranked pairs elections. Our results are in blue. Results marked by $\blacklozenge$ are due to Parkes and Xia [2012]; by $\blacklozenge$ due to Xia et al. [2009]; and by $\blacklozenge$ claimed to be in P by Menton and Singh [2012], but later stated as open [2013] (and omitted from their most recent arXiv version, v4, dated May 24, 2013), and re-established in Theorem 4. $\blacklozenge$ marks a result by Parkes and Xia [2012], extended by Gaspers et al. [2013]. The original proof of Menton and Singh [2013] for the result marked by $\blacklozenge$ is corrected in Section 3 and extended to the unique-winner model in Theorem 2. All results, except Schulze-DCDC (where the unique-winner model remains open), hold in both winner models.

For a given election \((C, V)\), we first introduce the weighted majority graph (WMG). A WMG for an election \((C, V)\) is a weighted directed graph \(G = (V, E, w)\), where \(V = C\) and \((c, d) \in E\) with weight \(w(c, d) = D_V(c, d)\) for each ordered pair of candidates \(c, d \in C\). Note that, since the votes are expressed as strict orders, we have \(D_V(c, d) = D_V(d, c)\). Therefore, we omit edges with a negative weight in depictions.

A voting rule \(r:\{(C, V)\mid (C, V)\text{ is an election}\} \to 2^C\) determines the set of winners of an election \((C, V)\). We focus on the voting rules Schulze and ranked pairs. A candidate \(c \in C\) is a Condorcet winner of an election \((C, V)\) if \(N_v(c, d) > N_v(d, c)\) holds for all \(d \in C\) with \(c \neq d\). A candidate \(c \in C\) is a weak Condorcet winner if \(N_v(c, d) \geq N_v(d, c)\) for all \(d \in C\) with \(c \neq d\). Note that there can be at most one Condorcet winner but possibly multiple weak Condorcet winners. Both Schulze and ranked pairs are Condorcet-consistent voting rules, meaning they choose the Condorcet winner whenever there is one.

2941
Schulze. Let the strength of a path $p$ ($\text{str}(p)$) be the weight of the weakest edge, i.e., the minimum weight, in a directed path $p$ between two candidates in the WMG. For each distinct pair of candidates $c, d \in C$, let the strength of the strongest path be $P(c, d) = \max \{\text{str}(p) \mid p$ is a path from $c$ to $d\}$. A candidate $c \in C$ is a Schulze winner of $(C, V)$ if $P(c, d) \geq P(d, c)$ for each $d \in C \setminus \{c\}$. To find these candidates one can build a second directed graph, which has an edge from $c$ to $d$ if and only if $P(c, d) > P(d, c)$. Each candidate with an in-degree of zero in this graph is a Schulze winner. There can be multiple Schulze winners.

Ranked pairs. For an election $(C, V)$, we first calculate $D_{V}(c, d)$ for all pairs of distinct candidates $c, d \in C$, and order the pairs by weight from highest to lowest, i.e., we order the values of $D_{V}(c, d)$. Now in each step, we consider the top pair $(c, d)$ of this weight order, which has not yet been considered. Following Parkes and Xia [2012], we break ties according to a fixed tie-breaking rule. We add an edge $(c, d)$ to a directed graph $G = (V', E)$ where $V' = C$, unless inserting this edge would create a cycle, in which case the pair (edge) is disregarded. When all pairs have been considered, the ranked pairs winner of $(C, V)$ (subject to the fixed tie-breaking) is the candidate corresponding to the source of $G$.

For examples of the Schulze and ranked pairs method we refer to the supplementary material. Note that Tideman [1987] originally gives an irresolute procedure and corresponding function for ranked pairs, which considers all possible ways of breaking ties and is NP-complete [Brill and Fischer, 2012]. Interestingly, the full ranking of candidates and winner determination including ties can be computed in polynomial time for Schulze [2023]. For our reductions, it is important that the procedures run in polynomial time (which we achieve by using a fixed tie-breaking scheme for ranked pairs), but it is not necessary to be highly efficient. Consequently, there may be other procedures that are faster (see, e.g., [Sornat et al., 2021] for Schulze) or handle ties in a different way for ranked pairs (see, e.g., [Brill and Fischer, 2012; Wang et al., 2019]).

We study various types of electoral control, starting with constructive control by deleting candidates (CCDC) as defined by Bartholdi et al. [1992] for a voting rule $\mathcal{E}$:

**$\mathcal{E}$-CONSTRUCTIVE CONTROL BY DELETING CANDIDATES**

Given: An election $(C, V)$, a distinguished candidate $p \in C$, and $\ell \in \mathbb{N}$.

Question: Is it possible to make $p$ the unique winner of the $\mathcal{E}$ election resulting from $(C, V)$ by deleting at most $\ell$ candidates?

In the setting of replacing candidates or voters [Loreggia et al., 2015; Erdélyi et al., 2021], the chair must not alter the size of the election and instead must add a candidate or voter for each one she deletes. Formally, in $\mathcal{E}$-CONSTRUCTIVE CONTROL BY REPLACING CANDIDATES ($\mathcal{E}$-CCRC) we are given two disjoint sets of candidates, $C$ and $D$, a list of votes over $C \cup D$, a distinguished candidate $p \in C$, and $\ell \in \mathbb{N}$, and we ask if it is possible to make $p$ the unique winner of the $\mathcal{E}$ election resulting from $(C, V)$ by replacing at most $\ell$ candidates $C' \subseteq C$ with candidates $D' \subseteq D$, where $|C'| = |D'|$.

We abbreviate multimode control problems in the obvious way; e.g., we use the shorthand $\mathcal{E}$-CCAC+DC+AV+DV+B for the above problem. Faliszewski et al. [2011] define a method to classify all $2^3 - 1$ variants of multimode control for a voting rule, called classification rule A. Using the classification rule A and the known results for adding and deleting candidates or voters and for bribery (see Table 1), it immediately follows that, except for Schulze-DCAC+DC, Schulze is resistant to any multimode attack. Since ranked pairs is resistant to all single-pronged attacks, it clearly also resists all combinations of multimode control.

In any instance of $\mathcal{E}$-EXACT CONSTRUCTIVE CONTROL BY AC+DC+AV+DV+B, it must hold that $|D'| = \ell_{AC}$. 

---

4In the original definition, the voting rule ranked pairs [Tideman, 1987] returns a complete ranking of the candidates. We use a slightly simplified definition of ranked pairs introduced by Berker et al. [2022] that only returns the winner of the election.
\[ C' = \ell_{DC}, \quad [U'] = \ell_{AV}, \quad [V'] = \ell_{DV}, \quad \text{and exactly } \ell_B \]

voters in \((V \setminus V') \cup U'\) are bribed. Each corresponding nonexact control problem polynomial-time Turing reduces to the exact control problem. The destructive variants \(\mathcal{E}-\text{DESTRUCTIVE CONTROL BY AC}+\text{DC}+\text{AV}+\text{DV}+\text{B}\) and \(\mathcal{E}-\text{EXACT DESTRUCTIVE CONTROL BY AC}+\text{DC}+\text{AV}+\text{DV}+\text{B}\) are defined analogously by asking whether it is possible to make \(p\) not a (unique) winner, and we again use the obvious shorthands. Sometimes, we exclude certain actions from multimode control, considering, e.g., only candidate control (\(\mathcal{E}\)-DCAC+DC) and omit the unneeded input parameters.

Note that we do not allow candidates in \(D\) or voters in \(U\) to be deleted, as we do not consider it realistic to remove a candidate or voter from an election right after adding. Even though this does not make a difference for the nonexact problems (as it is allowed to simply add or delete fewer candidates or voters), it may affect the result for exact multimode control as shown in the supplementary material.

3 Schulze Resists Constructive Control by Deleting Candidates

In this section, we prove the following result by describing and fixing a flaw in the proof of Menton and Singh [2013].

**Theorem 1.** Schulze-CCDC is NP-complete in the nonunique-winner model.

**Proof.** The proof of this result, due to Menton and Singh [2013, Thm. 2.2], shows a clever reduction from 3SAT, but it is technically flawed. We briefly present their reduction and give a counterexample showing that it is not correct.

In the 3-SATISFIABILITY problem (3SAT), we are given a set \(X\) of variables and a set \(C = \{C_l_1, \ldots, C_l_k\}\) of clauses over \(X\), each having exactly three literals, and we ask whether there is a satisfying assignment for \(\varphi\), where \(\varphi\) is the conjunction of all clauses \(C_l_i \in C\). Given a 3SAT instance \((X, C)\), Menton and Singh [2013] construct a Schulze-CCDC instance \(((X, V'), p, k)\) as follows. The set of candidates \(C\) contains \(k+1\) clause candidates \(C_{1}, \ldots, C_{k+1}\) for each clause \(C_l_i \in C\), three literal candidates \(x_{i,j}^l, x_{i,j}^r, x_{i,j}^s\) for each clause \(C_l_i\), where \(x_{i,j}^l\) is the \(j\)th literal in clause \(C_l_i\), \(k+1\) negation candidates \(n_{i,j,m,n}^1, \ldots, n_{i,j,m,n}^{k+1}\) for each pair of literals \(x_{i,j}^l, x_{i,j}^m\), and the distinguished candidate \(p\) and an additional candidate \(a\). Let \(C_i = \{c_{1}, \ldots, c_{k+1}\}\) be the set of all clause candidates for clause \(C_l_i \in C\) and let \(K = \bigcup_{i=1}^{k+1} C_i\) be the set of all clause candidates. Let \(L_i = \{x_{i,j}^l, x_{i,j}^r, x_{i,j}^s\}\) be the set of literal candidates for the clause \(C_l_i\) and let \(L = \bigcup_{i=1}^{k+1} L_i\) be the set of all literal candidates. Let \(N_{ijmn} = \{n_{ijmn}^1, \ldots, n_{ijmn}^{k+1}\}\) be the set of negation candidates for the literals \(x_{i,j}^l, x_{i,j}^m\) that are a negation of each other, and let \(N\) be the set of all such negation candidates. For a positive integer \(z\), we write \([z] = \{1, \ldots, z\}\) as a shorthand.

Menton and Singh [2013] define the following list of votes \(V'\) (which we will change later to fix the proof):

<table>
<thead>
<tr>
<th>#</th>
<th>preferences</th>
<th>for each</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A)</td>
<td>1 (W(x_{i,j}^l, x_{i,j}^m))</td>
<td>(i \in [k], j \in [k+1])</td>
</tr>
<tr>
<td>(B)</td>
<td>1 (W(x_{i,j}^l, x_{i,j}^r))</td>
<td>(i \in [k])</td>
</tr>
<tr>
<td>(C)</td>
<td>1 (W(x_{i,j}^l, x_{i,j}^s))</td>
<td>(i \in [k])</td>
</tr>
<tr>
<td>(D)</td>
<td>1 (W(x_{l,i}^l, p))</td>
<td>(i \in [k])</td>
</tr>
<tr>
<td>(E)</td>
<td>1 (W(a, x))</td>
<td>(x \in L)</td>
</tr>
<tr>
<td>(F)</td>
<td>1 (W(p, a))</td>
<td></td>
</tr>
<tr>
<td>(G)</td>
<td>1 (x_{l,i}^l, n_{ijmn}^l)</td>
<td>(l \in [k+1]) where (x_{i,j}^m) is the negation of (x_{l,i}^l)</td>
</tr>
<tr>
<td>(H)</td>
<td>1 (W(n, p))</td>
<td>(n \in N)</td>
</tr>
</tbody>
</table>

The deletion limit is \(k\), the number of clauses. Menton and Singh [2013] argue that \(p\) can be made a Schulze winner by deleting at most \(k\) candidates from \(C\) if and only if there is a truth assignment that makes the given 3SAT instance true.

We now briefly present our counterexample, where we map a yes-instance of 3SAT to a no-instance of Schulze-CCDC. Let \((X, C)\) be our given 3SAT instance, with \(X = \{x_{1}, x_{2}, x_{3}\}\) and \(C = \{(x_{1} \lor x_{2} \lor \neg x_{3}), (\neg x_{1} \lor x_{2} \lor x_{3})\}\), i.e., we consider the CNF formula

\[ \varphi = (x_{1} \lor x_{2} \lor \neg x_{3}) \land (\neg x_{1} \lor x_{2} \lor x_{3}). \]

A detailed description of this counterexample is given in the supplementary material. Their reduction is quite clever, but unfortunately wrong, as shown by the counterexample. However, by modifying it appropriately, we can ensure that \(p\) can indeed be made a Schulze winner of the election by deleting at most \(k\) candidates if and only if \((X, C)\) is a yes-instance of 3SAT.

For our modifications, it is only necessary to change the list of votes. For votes (A) through (F) we adjust the number of votes to 2. Additionally, we add 1 vote \(W(a, c)\) for each \(c \in K\). The graph in Figure 1 shows the weighted majority graph for the 3SAT instance from our counterexample adapted to the new reduction. We claim that \((X, C)\) is a yes-instance of 3SAT if and only if \(((C, V'), p, k)\) is a yes-instance of Schulze-CCDC in the nonunique-winner model.

From left to right, let \((X, C)\) be a yes-instance of 3SAT. Since we have a yes-instance of 3SAT, we have a truth assignment that makes at least one literal in each clause \(C_{l_i} \in C\) true. We claim that \(p\) can be made a Schulze winner by deleting one literal candidate corresponding to some true literal for each clause. The only path with weight four from a clause candidate \(c_{l_i} \in K\) to \(p\) is through the literal candidates: from \(x_{l_i}^l\) via \(x_{l_i}^r\) to \(x_{l_i}^s\). Since we deleted one literal candidate for each clause, there no longer exists a weight-4 path from a clause candidate to \(p\) and \(P(p, c) = 2 \geq P(c, p)\) for \(c \in K\). For each \(c \in L \cup \{a\}\), we have \(P(p, c) = 4 \geq P(c, p)\). Since we deleted only literal candidates where the corresponding literal was assigned to be true, we never have the case that we deleted two literal candidates \(x_{l_i}^l, x_{l_i}^m\) which negate each.
other. Thus we still have a weight-2 path from \( p \) to each nega-
tion candidate and \( P(p, c) = 2 = P(c, p) \) for each \( c \in N \).
It follows that \( (\{C, V\}, p, k) \) is a yes-instance of Schulze-
CCDC in the nonunique-winner model.

From right to left, let \((X, Cl)\) be a no-instance of \( 3\text{SAT} \).
Thus, for each assignment of the literals, there exists a clause
which is false. To ensure that \( p \) is a winner of the election,
it is necessary that \( P(p, c) \geq P(c, p) \) for each \( c \in C \setminus \{ p \} \).
Since \( P(p, c) = 2 < 4 = P(c, p) \) for each \( c \in K \), we have to
destroy each path of weight greater than two from the clause
candidates to \( p \), in particular the path through the literal
candidates \( x_i^j \in C_i, j \in \{1, 2, 3\} \). Due to the deletion limit
\( k = |Cl| \), it is necessary to delete one literal candidate for
each clause. Consider any subset of literals of size \( k \) such
that for each clause one literal is contained in the set. If all of
these subsets contained at least two literals, \( x_i^j \) and \( x_m^n \), that
negate each other, i.e., if the \( 3\text{SAT} \) formula were not satis-
ifiable, by deleting the corresponding two literal candidates, we
would no longer have a path from \( p \) to the negation candidates
\( n \in N_{ijmn} \). That is, it would hold that \( P(p, n) < P(n, p) \)
and it would be impossible to make \( p \) a Schulze winner by
deleting at most \( k \) candidates. Thus, since we have a yes-
instance of Schulze-CCDC, there must exist at least one subset
of literals, such that for each clause one literal is contained in
the set, i.e., set to true and we have \( 3\text{SAT} \)-yes.

Finally, it is easy to see that Schulze-CCDC is in \( \NP \). \( \square \)

Previously, Schulze-CCDC was only studied in the
nonunique-winner model. By slightly adapting the previous
construction (see the supplementary material), we can now
also show \( \NP \)-completeness in the unique-winner model.

\textbf{Theorem 2.} Schulze-CCDC is \( \NP \)-complete in the unique-
winner model.

\section{Destructive Control by Deleting Candidates
and Variants Thereof}

In this section, we examine a peculiarity of standard de-
structive control by deleting candidates for Schulze elections,
which allows us to reduce the number of candidates we need
to consider for deletion. Using this approach, we are able to
re-establish the result that DCDC is polynomial-time solv-
able for Schulze elections, which Menton and Singh claimed
in an early version (v1) of their arXiv preprint [Menton and
Singh, 2012]. However, Menton and Singh removed this result
(and the corresponding result for Schulze-DCAC) from all subsequent versions of the arXiv preprint and from their
IJCAI 2013 paper [Menton and Singh, 2013], stating these
two control problems as open. Additionally, we consider the
relationship of variations of destructive control by deleting
candidates to variations of \( s \)-\( t \) vertex cut. All omitted proofs
can be found in the supplementary material.

We first introduce a variant of \( s \)-\( t \) vertex cut defined by
Menton and Singh [2013]: \textsc{Path-Preserving Vertex Cut}. Recall that for destructive control in Schulze elec-
tions to be successful, one candidate must be boosted to beat
the despised candidate and that by deleting candidates we
can only lower the strength of the strongest path but can
never increase it. Hence, we need to cut paths such that
the best remaining path to the despised candidate is stronger
than all paths from the despised to our chosen boosted can-
date. Menton and Singh [2013] capture this notion of path-
preserving vertex cut by defining the problem as essentially
asking whether there exists an \( s \)-\( t \) vertex cut (destroying the
stronger paths from \( s \) to \( t \) of size at most \( k \), while at least
one path from \( t \) to \( s \) must remain intact. When searching for
candidates to delete (so as to boost some candidate to beat the
despised candidate), it is useful if we can reduce the search
space. We will at times refer to these candidates as \textit{rivals of the
despised candidate}. For the standard control problem we
show that this is indeed possible and the chair can limit the
search to the in-neighborhood of the boosted candidate.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{figure1.png}
\caption{WMG corresponding to the election constructed from the
\( 3\text{SAT} \) instance from our counterexample with the new reduction.
Dashed edges have weight two and drawn edges have weight four.}
\end{figure}
Theorem 3. If destructive control by deleting candidates is possible for a given Schulze election, then there exists some candidate \(c \in C\) who can beat the despaired candidate \(w\) by only deleting candidates who directly beat \(c\) (i.e., are in-neighbors of \(c\) in the WMG).\(^6\)

Proof. Let \(c \in C\) be some candidate where \(P(w, c) \geq P(c, w)\), i.e., \(w\) has a stronger (or equally strong) path to \(c\) than the other way round. Assume that \(c\) can beat \(w\) by deleting the minimal number of candidates needed (with regards to the number necessary for making other candidates beat \(w\)). We claim that either those deleted candidates are direct neighbors of \(c\), or there exists some other candidate \(c^* \in C\) for which we can reach the same goal by deleting equally many or even fewer candidates in the neighborhood of \(c^*\).

First, we define some notation. Let \(\text{Del}^c_w\) be the minimal number of removed candidates needed to make \(c\) beat \(w\). Note that these candidates form a path-preserving vertex cut. We say that \(x\) is before \(z\) if \(x\) is closer to \(w\) than \(z\).

Let \(N^+(c)\) be the in-neighborhood of a candidate \(c\), i.e., \(N^+(c)\) contains all candidates with a direct edge to \(c\). Finally, we define \(\text{Ind}^c_w\) as the candidates belonging to the connected component of \(c\) as induced by the vertex cut \(\text{Del}^c_w\). Intuitively, \(\text{Ind}^c_w\) contains all candidates on stronger paths from \(w\) to \(c\) that are broken by deleting candidates and where the cut is before \(c\).

Clearly, any \(z^* \in \text{Ind}^c_w\) also beats \(w\) as first, \(P(z^*, c) \geq P(c, w)\) and thus \(P(z^*, w) \geq P(c, w)\), and second, no path from \(w\) to \(z^*\) with strength greater than \(P(c, w)\) can exist. It follows that \(|\text{Del}^c_w| \leq |\text{Del}^c_w|\).

We distinguish two cases.

Case 1: \(\text{Ind}^c_w = \emptyset\). Since \(\text{Del}^c_w\) is a minimal cut, we have \(\text{Del}^c_w \subseteq N^+(c)\).

Case 2: \(\text{Ind}^c_w \neq \emptyset\). Let \(F = \{ f \in \text{Ind}^c_w \mid N^+(f) \cap \text{Del}^c_w \neq \emptyset\}\) be the set of all candidates in the connected component of \(c\), where we deleted some candidates in the in-neighborhood of \(c\). On the one hand, if \(|F| = 1\) we have a candidate \(f \in F\) who also beats \(w\) by deleting \(\text{Del}^c_w\). Since \(\text{Del}^c_w\) is minimal, it follows that \(\text{Del}^c_w = \text{Del}^c_w\), and therefore, for a successful control action against \(w\), it suffices to delete from \(N^+(f)\).

On the other hand, if \(|F| > 1\), then \(N^+(f) \cap \text{Del}^c_w = N^+(g) \cap \text{Del}^c_w\) for all \(f, g \in F\). For a contradiction, assume there are two candidates \(f, g \in F\) who do not share the same in-neighbors in \(\text{Del}^c_w\). Deleting either \(N^+(f) \cap \text{Del}^c_w\) or \(N^+(g) \cap \text{Del}^c_w\) is sufficient to make the respective candidate beat \(w\). Since \(N^+(f) \cap \text{Del}^c_w \neq N^+(g) \cap \text{Del}^c_w\), we have a contradiction to \(\text{Del}^c_w\) being minimal. \(\blacksquare\)

This result can be used to design an algorithm for Schulze-DCDC in the unique-winner model, which runs in polynomial time. Unfortunately, we cannot easily transfer our algorithm to work in the unique-winner model. The complete algorithm, proof of correctness (based on Theorem 3), and further explanation as well as an example of why the algorithm cannot solve the unique-winner model can be found in the supplementary material. Intuitively, for a despaired candidate \(d\), the algorithm works by considering every candidate \(c \in C \setminus \{d\}\) as a possible rival of \(d\) and testing whether \(c\) is successful by considering the part of the in-neighborhood of \(c\) that intersects the stronger paths from \(d\) to \(c\) for deletion.

Theorem 4. In the nonunique-winner model, Schulze-DCDC is solvable in polynomial time.

For variants of destructive control, we need to encode the restriction, which is imposed on the deletion set, into the vertex cut, essentially creating corresponding new PATH-PRESERVING VERTEX CUT decision problems. There are several natural restrictions one may apply to Schulze-DCDC, e.g., some candidates may be protected from deletion or labeled candidates must be deleted together. We give examples and prove the relationship to vertex cut for some of these variants in the supplementary material.

5 Exact Multimode Control and Control by Replacing

Now we turn to exact multimode control and control by replacing candidates or voters. In the former control type, the chair must adhere to alter an exact number of candidates or voters or both. In the latter, the chair must add the same number of either candidates or voters as previously have been deleted, i.e., must replace them. Lorregia et al. [2015] showed that any voting rule that is resistant to constructive control by deleting candidates and satisfies insensitivity to bottom-ranked candidates (IBC) [Lang et al., 2013] is also resistant to constructive control by replacing candidates. A voting rule \(\mathcal{E}\) is said to be insensitive to bottom-ranked candidates if, given an election \((C, V)\) and a new candidate \(x\), the elections \((C, V)\) and \((C \cup \{x\}, V^x)\), where \(V^x\) is the list of votes obtained by adding \(x\) as the least preferred alternative to each vote in \(V\), have the same winners under \(\mathcal{E}\). We extend their result to also apply to \(\mathcal{E}\)-ECCAC+DC and \(\mathcal{E}\)-ECCR.

Table 2 provides an overview of our results. All (full) proofs omitted in this section can be found in the supplementary material.

<table>
<thead>
<tr>
<th>Voting Rule</th>
<th>Schulze</th>
<th>Ranked pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exact AC+DC</td>
<td>NP-c.(^\bullet)</td>
<td>NP-c.(^\diamond)</td>
</tr>
<tr>
<td>Exact RC</td>
<td>NP-c.(^\bullet)</td>
<td>NP-c.(^\diamond)</td>
</tr>
<tr>
<td>Exact AV+DV</td>
<td>NP-c.(^\ast) NP-c.(^\ast)</td>
<td>NP-c.(^\circ) NP-c.(^\circ)</td>
</tr>
<tr>
<td>Exact RV</td>
<td>NP-c.(^\ast) NP-c.(^\ast)</td>
<td>NP-c.(^\circ) NP-c.(^\circ)</td>
</tr>
</tbody>
</table>

Table 2: Overview of complexity results for exact multimode control and control by replacing in both winner models. Results marked by \(\bullet\) can be found in Corollary 2 by \(\diamond\) in Corollary 3, by \(\ast\) in Corollary 1, by \(\circ\) in Theorem 5 and by \(^\circ\) in Theorem 6.

\(^6\)This result can be extended to any pair of candidates \(c, w \in C\) where \(P(w, c) \geq P(c, w)\) and we have a yes-instance of PATH-PRESERVING VERTEX CUT for the graph induced by the paths between these two candidates.
Let \( E \subset C \subset W \) with \( | C | = k \). Define \( D = \{ d_1, \ldots, d_{|C|} \}, \) \( \ell_{DC} = \ell_{RC} = k \), and set \( \ell_{AC} \in \mathbb{N} \) arbitrarily. Let \( V' = v \times D \) for every \( v \in V \), i.e., add all candidates from \( X \) at the bottom of every vote and then add all candidates from \( D \) at the bottom of those votes. Construct an instance \( (C', D, V', p, \ell_{AC}, \ell_{DC}) \) of \( \mathcal{E} \)-ECCAC+DC and an instance \( (C', V') \) of \( \mathcal{E} \)-ECCRRC.

Assume we have a yes-instance of \( \mathcal{E} \)-CCDC. Then there exists a set \( C_d \subset C \) with \( |C_d| \leq \ell_{DC} \) such that \( D \) wins the election \( (C \setminus C_d, V) \). Delete all candidates in \( C_d \) and some candidates in \( X_d \subset X \) such that exactly \( \ell_{DC} \) candidates were deleted and add \( \ell_{AC} \) arbitrary candidates \( D_a \) to the election. Since \( E \) is IBC, candidate \( p \) is a winner of the election \( (C' \cup D_a \setminus (C_d \cup X_d), V') \) and \( \mathcal{E} \)-ECCAC+DC is also a yes-instance.

Now assume, we have a no-instance of \( \mathcal{E} \)-CCDC. Then there is no set \( C_d \subset C \) with \( |C_d| \leq \ell_{DC} \) such that \( D \) wins the election \( (C \setminus C_d, V) \). Since \( E \) is IBC, deleting any set of candidates from \( X \) or adding any candidate from \( D \) has no influence on the winners of the election. Thus, to make \( p \) an \( \mathcal{E} \) winner, we need to find a set of at most \( \ell_{DC} \) candidates in \( C \) to delete, which is impossible as \( \mathcal{E} \)-CCDC is a no-instance.

The argument for \( \mathcal{E} \)-ECCRRC is analogous. \( \square \)

**Lemma 2.** Let \( E \) be a voting rule that satisfies IBC. In both the unique- and nonunique-winner model, if \( \mathcal{E} \)-DCDC is \( \mathcal{NP} \)-hard, then so are \( \mathcal{E} \)-EDCAC+DC and \( \mathcal{E} \)-EDCRRC.

The same construction and proof idea used in the proof of Lemma 1 works for Lemma 2 as well.

**Lemma 3.** Schulze and ranked pairs are insensitive to bottom-ranked candidates.

**Corollary 1.** In both the unique- and nonunique-winner model, Schulze-CCRC and Ranked-Pairs-CCRC are \( \mathcal{NP} \)-complete.

**Corollary 2.** In both the unique- and nonunique-winner model, Schulze-ECCAC+DC, Schulze-ECCRRC, Ranked-Pairs-ECCAC+DC, and Ranked-Pairs-ECCRRC are \( \mathcal{NP} \)-complete.

**Corollary 3.** In both the unique- and nonunique-winner model, Ranked-Pairs-EDCAC+DC, Ranked-Pairs-EDCRRC, and Ranked-Pairs-DCRCR are \( \mathcal{NP} \)-complete.

Schulze and ranked pairs are equally resistant to all constructive and destructive control actions of the voter list considered in this paper.

**Theorem 5.** In both the unique- and nonunique-winner model, Schulze-ECCAC+DV, Schulze-CCRV, Schulze-EDCAC+DV, and Schulze-DCRC are \( \mathcal{NP} \)-complete.

**Proof sketch.** We reduce from **Restricted Exact Cover By 3-Sets** (RX3C) [Gonzalez, 1985]: Given a set \( B = \{ b_1, \ldots, b_{3s} \} \) with \( s \geq 1 \) and a list \( S = \{ S_1, \ldots, S_{3s} \} \), where \( S_1 = \{ b_1, b_2, b_3 \} \) and \( S_i \subset B \) for all \( S_i \in S \) and each \( b_j \) is contained in exactly three sets \( S_i \in S \). Does there exist an exact cover, i.e., a sublist \( S' \subset S \) such that each \( b_i \in B \) occurs in exactly one \( S_i \in S' \)?

Let \( \ell_{AV} = \ell_{DV} = s \) for the Schulze-ECCAC+DV and Schulze-EDCAC+DV instances we construct, and let \( \ell_{RV} = s \) for the Schulze-CCRV and Schulze-DCRV instances. Further, let \( L \gg s \) be a constant much greater than \( s \). From \( (B, S) \) we construct an election \( (C, V) \) as depicted in Figure 2. Note that candidate \( w \) is the unique winner of the election. The list of additional votes \( U \) contains one vote

\[ S_i \setminus (B \setminus S_i) \quad \text{for each } S_i \in S. \]

Let \( p \) be the distinguished candidate for the constructive case and \( w \) be the despised candidate for the destructive case.

By adapting the above construction, we obtain the same results for ranked pairs.

**Theorem 6.** In both the unique- and nonunique-winner model, Ranked-Pairs-ECCAV+DV, Ranked-Pairs-CCRV, Ranked-Pairs-EDCAV+DV, and Ranked-Pairs-DCRV are \( \mathcal{NP} \)-complete.

### 6 Conclusion and Future Work

We have studied electoral control for Schulze and ranked pairs elections: After fixing a flaw in the proof of Menton and Singh [2013, Theorem 2.2] for Schulze-CCDC and extending the new construction to the unique-winner model (Section 3), we have studied variants of \( s \)-t vertex cuts in graphs that are related to destructive control by deleting candidates in Schulze elections, and have re-established a corresponding vulnerability result (Section 4). Finally, in Section 5, we established a number of resistance results for control by replacing candidates or voters and exact multimode control. Hence, our work establishes both polynomial-time algorithms and \( \mathcal{NP} \)-completeness results. Tables 1 and 2 provide a summary of our results in the context of related known results.

However, multiple variants of destructive control of the candidate set (see Section 4), such as destructive control by adding candidates both in the unique- and nonunique-winner model and by deleting candidates in the unique-winner model, remain open for Schulze elections. Lastly, to the best of our knowledge, most cases of control by partition are yet to be solved for ranked pairs elections.

---

\[ \text{The precise value of } L \text{ is not important; all that matters is that when used as an edge weight in a WMG, } L \text{ is large enough, such that any edge changed by the control actions must not change in direction, i.e., the sign of the edge weight must not flip. Recall that the strength of a path in a WMG is specified as the weight of the weakest edge on the path.} \]
**Ethical Statement**

The subject of this paper—electoral control—is an ethically sensitive topic. The goal of our work is not to help strategic parties to control elections but to ‘audit’ some voting rules with respect to their vulnerability to electoral control. In particular, our resistance results are clearly helpful to society since they may help protect society from attacks against elections, and our vulnerability results are clearly helpful to society since they can be used to avoid voting rules that are vulnerable to certain types of electoral control that realistically may occur in the context of an election. Hence, the purpose of our research is to provide detailed information on the societal impact of voting rules and to ensure transparency.

**Acknowledgments**

We are grateful to Ulle Endriss, Piotr Faliszewski, Edith Hemaspaandra, Lane A. Hemaspaandra, and Jérôme Lang for their ethical advice. We thank the anonymous reviewers for helpful comments. This work was supported in part by DFG research grant RO-1202/21-1.

**References**


[Hemaspaandra et al., 2013] Lane A. Hemaspaandra, Raman Lavaee, and Curtis Menton. Schulze and ranked-pairs...


