ECAI 2014
T. Schaub et al. (Eds.)

doi:10.3233/978-1-61499-419-0-411

Controlling Two-Stage Voting Rules

Jiong Guo [1] and **Yash Raj Shrestha** [2]

Abstract. We study the computational complexity of control problems for two-stage voting rules. An example of a two-stage voting rule is the Black's procedure. The first stage of the Black's procedure selects the Condorcet winner if one exists; otherwise, in the second stage the Borda winner is selected. The computational complexity of the manipulation problem of two-stage voting rules has recently been studied by Narodytska and Walsh [20] and Fitzsimmons et al. [14]. Extending their work, we consider the control problems for similar scenarios, focusing on constructive control by adding or deleting votes, denoted as CCAV and CCDV, respectively.

Let X be the voting rule applied in the first stage and Y the one in the second stage. As for the manipulation problem shown in [20, 14], we prove that there is basically no connection between the complexity of CCAV and CCDV for X or Y and the complexity of CCAV and CCDV for the two-stage election X THEN Y: CCAV and CCDV for X THEN Y could be NP-hard, while both problems are polynomial-time solvable for X and Y. On the other hand, combining two rules X and Y, both with NP-hard CCAV and CCDV, could lead to a two-stage election, where both CCAV and CCDV become polynomial-time solvable. Hereby, we also achieve some complexity results for the special case X THEN X. In addition, we show that, compared to the manipulation problem, the control problems for two-stage elections admit more diverse behaviors concerning their complexity. For example, there exist rules X and Y, for each of which CCAV and CCDV have the same complexity, but CCAV and CCDV behave differently for X THEN Y.

1 Introduction

There exist several voting procedures involving two or more stages. For example, the Black's procedure is a two-stage voting rule, where in the first stage the Condorcet winner is elected if one exists; otherwise, it moves to the second stage which elects the Borda winner [13]. As a real-world example, the French presidential elections use a two-stage runoff voting system [20]. If there is a majority winner in the first stage, then this candidate is the overall winner; otherwise, the second stage applies a runoff vote between the two candidates with the most votes in the first stage. As mentioned in [20], these two-stage voting rules can inherit a number of attractive axiomatic properties from the rules applied in the individual stages. For example, the Black's procedure inherits the Condorcet consistency from its first stage, and properties like monotonicity, participation and the Condorcet loser property from its second stage. Inheriting such properties could be one attractive feature of voting rules involving more than one stage. On the negative side, some less desirable property of the rules of the individual stages could also affect the overall two-stage rules. For example, with single-peaked votes, many types of control and manipulation problems are polynomial-time solvable for the Black's procedure [3]. This polynomial cost is inherited from the first stage of the rule, which selects the Condorcet winner (which must exist with single-peaked votes). Such vulnerability to manipulation control is considered as an undesirable property of voting rules.

Recently, two-stage voting rules have been extensively studied. Initialized by studies in economics, multi-stage elections and runoffs have become more and more influential in computational social choice during the past decade [6, 4]. Particularly, some interesting work concerning strategic attacks on two-stage voting has been done by Narodytska and Walsh [20]. They focused on election systems of the form X THEN Y, i.e., an initial-round election under voting rule X, after which if there are multiple winners, then only those winners enter a runoff election under voting rule Y, with the initial votes now restricted to these winners. Hereby, the manipulation problem asks whether a given manipulation coalition can vote in such a way to make a distinguished candidate win (namely, win in the initial round if there is a unique winner in the initial round, or if not, then be a winner of the runoff). They mainly studied the issue of how the manipulation complexity of the rules X and Y affects the manipulation complexity of X THEN Y. Using the tools from classical complexity theory where polynomial-time solvability (P for short) is considered as being easy and NP-hardness (NP-h for short) as being hard, they showed that every possible combination of these manipulation complexities can be achieved for X, Y and X THEN Y. Fitzsimmons et al. [14] studied the complexity of the manipulation problem of the special two-stage voting rules X THEN X. They also considered the case, where revoting is allowed in the second stage. There are real-world examples of such same-system runoff elections as well. For instance, in North Carolina and many districts of California, the election law specifies that if there are two or more candidates tied in the initial plurality election, a plurality runoff election is held among these winners [14].

Elkind and Lipmaa [6] considered manipulating the multiple-state elections with the same rule applying in all stages. However, their model, in contrast to the one used by Narodytska and Walsh [20] and Fitzsimmons et al. [14] and also in this paper, is based on removing only the least successful candidate after each round. Recently, Bag et al. [1] and Davies et al. [5] also studied the manipulation of the multiple-stage model with removing only the weakest candidates sequentially. Related to the model by Elkind and Lipmaa [6], Conitzer and Sandholm [4] introduced the "universal tweaks", and showed that adding one so-called pairwise CUP-like "pre-round", which cuts out about half of the candidates, can tremendously boost the manipulation complexity for a broad range of election systems.

[1] Cluster of Excellence, Universitäte des Saarlandes, Germany, email: jguo@mmci.uni-saarland.de
[2] Cluster of Excellence, Universitäte des Saarlandes, Germany, email: yashraj@mmci.uni-saarland.de

This paper extends the work of Narodytska and Walsh [20] and Fitzsimmons et al. [14] to the control problems. We study the complexity of the arguably most important types of control in two-stage elections, that is, adding and deleting votes. Control by deleting (adding) votes asks whether in a given election a candidate who is preferred by the controlling agent can be made to win by deleting (adding) at most a certain number of votes (at most a certain number of votes from potential additional votes). These control types model strategic behaviors that arise in many electoral settings ranging from human to electronic. These issues are often faced by people seeking to steer an election, such as experts doing campaign management, and deciding for example which k people to offer rides to the polls [11].

The study of computational complexity of control behaviors was initialized in 1992 in the seminal paper by Bartholdi, Tovey and Trick [2], who considered constructive control by adding/deleting/partitioning candidates/votes under the plurality and Condorcet rules. A major motivation for the study of control was to obtain "hardness" results based on classical complexity theory, that is, results showing that determining optimal strategy for various control attacks is computationally infeasible. This research direction was continued by Hemaspaandra, Hemaspaandra, and Rothe [17], who studied destructive control attacks which prevent a particular candidate from being a winner through various control actions. Since then, many studies have been conducted on electoral control problems in various settings and for many different rules; we refer the readers to the survey [9]. Some recent research, not covered in that survey, considered control problems for the k-approval rules [18], for the Bucklin's rule (and for the fallback rule and its extension for truncated votes) [7, 8], for the maximin rule [10], for the range voting [19] and for the Schultze's rule and the ranked pairs rule [21]. Faliszewski et al. [11] have recently studied the computational complexity of weighted electoral control.

We focus on the constructive control problems by adding/deleting votes for the X THEN Y elections. Here, both X and Y could be same or different. In a similar fashion as shown by Narodytska and Walsh [20] and Fitzsimmons et al. [14] for the manipulation problem, we prove that there is no general relation between the control complexity of X THEN Y and the control complexity of the rules X and Y. More precisely, depending on X and Y, every combination of polynomial-time solvability and NP-hardness could be possible for the control complexities of X, Y and X THEN Y, as shown in Table 1. Note that, for X and Y being natural concrete voting rules, we witness a "complexity increment" of control problems by combining X and Y. In other words, a control problem of X THEN Y with two natural voting rules X and Y is polynomial-time solvable, only if the same control problem is polynomial-time solvable for both X and Y. We can only prove, for two artificially created rules X and Y, that even if a control problem is NP-hard for both X and Y, one can solve the same control problem in polynomial time for the overall two-stage election. As a byproduct, we also achieve some complexity results for the case $X = Y$ as shown in Table 2, complementing the work by Fitzsimmons et al. [14] for the manipulation problems for X THEN X. Hereby, we examine several combinations of prominent voting rules, for instance, r-Approval THEN r-Approval, Veto THEN r-Approval, r-Approval THEN CONDORCET, etc. Of particular interest is the two-stage rule Veto THEN Veto. With all other examples, we can observe an identical complexity behavior of control by adding votes and control by deleting votes for X THEN Y, if both control problems have the same complexity for X as well as for Y. However, for Veto THEN Veto, although both adding votes and deleting votes versions are polynomial-time solvable for Veto voting, control by deleting votes remains polynomial-time solvable for Veto THEN Veto, but control by adding votes turns out to be NP-hard. It is open whether there are other rule combinations with this property. By this example, we believe that the control problems, compared to the manipulation problems, seemingly offer more research opportunities with two-stage rules.

Preliminaries We take an election to be a pair $E = (C, V)$, where C is a set of candidates and V is a set of votes. Each vote represents the "preference" of the corresponding voter over C. A *preference* is a total, linear order that ranks the candidates from the most preferred one to the least preferred one. For example, if $C = \{a, b, c\}$ and some voter likes a best, then b, and then c, then his or her preference is $a \succ b \succ c$.

A voting rule is a function R that given an election $E = (C, V)$ returns a subset $R(E) \subseteq C$ of the candidates, that are said to win the election. Typically, we expect rules to have a unique winner, but sometimes ties can happen. In the initial round of the X THEN Y election, we assume a **non-unique winner** model where all tied-for-winning candidates are called winners. However, the second round should provide a **unique winner**.

In this paper, we consider the following voting rules. An m-candidate *scoring rule* is defined through a nonincreasing vector $\alpha = (\alpha_1, \ldots, \alpha_m)$ of nonnegative integers. A candidate $c \in C$ is given α_i points from each voter that ranks c in the i^{th} position of his preference. The candidate(s) with the maximum score are the winners. Many election rules can be considered as scoring rules, for instance, Veto and Approval-based systems. The *Veto* rule has the vector $(1, 1, \ldots, 1, 0)$. In Veto voting, a candidate c is said to have k vetoes, if c is the last preferred candidate in k votes. In *k-Approval* voting, each voter gives one point to each of his or her k most favorite candidates. Given an election $E = (C, V)$, a candidate c is a Condorcet (or weak Condorcet) winner if for every other candidate $d \in C \setminus \{c\}$, it holds that more than half (or at least half) of the voters prefer c to d. Note that it is possible that there is no (weak) Condorcet winner in a given election.

We consider a general class of two-stage voting rules. Given voting rules X and Y, the rule X THEN Y applies the voting rule Y to the profile constructed by eliminating all but the winning candidates from the voting rule X. Both X and Y can themselves be two-stage voting rules. Moreover, X and Y could be same or different voting rules.

Definition 1. *In X THEN Y elections, for the problem of constructive control by adding votes (denoted as CCAV) and the problem of constructive control by deleting votes (denoted as CCDV), the input contains a tuple $(E = (C, V), p, k)$ where C is a set of candidates, V is a collection of registered votes (with preferences over C), $p \in C$ is a preferred candidate, and k a nonnegative integer. In CCAV we also have an additional collection W of unregistered votes (with preferences over C). The questions are:*

***CCAV.** Is there a subcollection W' of W with at most k votes, such that $p \in R(C, V + W')$?*

***CCDV.** Is there a subcollection V' of V with at most k votes, such that $p \in R(C, V - V')$?*

In the above definition, "$+$" and "$-$" represent adding and deleting a set of votes, respectively.

Most NP-hardness reductions are from the Exact Cover by 3-Sets (X3C) problem:

Table 1. CCAV and CCDV in X THEN Y

X		Y		X THEN Y		Evidence
CCAV	CCDV	CCAV	CCDV	CCAV	CCDV	
P	P	P	P	P	P	Theorem 1
P	P	P	P	NP-h	NP-h	Theorem 2
P	P	P	P	NP-h	P	Theorem 3
P	P	NP-h	NP-h	P	P	Theorem 6
P	P	NP-h	NP-h	NP-h	NP-h	Theorem 4
NP-h	NP-h	P	P	P	P	Theorem 7
NP-h	NP-h	P	P	NP-h	NP-h	Theorem 5
NP-h	NP-h	NP-h	NP-h	P	P	Theorem 11
NP-h	NP-h	NP-h	NP-h	NP-h	NP-h	Theorem 9

Table 2. CCAV and CCDV in X THEN X

X		X THEN X		Evidence
CCAV	CCDV	CCAV	CCDV	
P	P	P	P	Corollary 1
P	P	NP-h	NP-h	Theorem 2
P	P	NP-h	P	Theorem 3
NP-h	NP-h	NP-h	NP-h	Theorem 10

Input: A set $\mathcal{B} = \{B_1, \ldots, B_{3m}\}$, and a collection $\mathcal{S} = \{S_1, \ldots, S_n\}$ of 3-element subsets of $\mathcal{B}$
Question: Does $\mathcal{S}$ have an exact cover S' for $\mathcal{B}$, i.e., a subcollection S' of $\mathcal{S}$ such that every element of $\mathcal{B}$ occurs in exactly one subset of S' ?

RESTRICTED EXACT 3-SET COVER (RX3C) is defined similarly to X3C, with the additional condition that each element in $\mathcal{B}$ appears in exactly three subsets of $\mathcal{S}$. RX3C is NP-complete [16].

Another problem we use is NP-hard VERTEX COVER on 3- regular graphs [15] which is defined as follows:

Input: A 3-regular graph $G' = (V', E')$ and an integer t.
Question: Does there exist a subset $S \subseteq V'$ of size at most t such that each edge in E' has at least one endpoint in S ?

A 3-regular graph is a graph where exactly three edges are incident to every vertex. We also use NP-hard INDEPENDENT SET in 3-regular graphs [16] which is defined as follows:

Input: A 3-regular graph $G = (V', E')$ and an integer k
Question: Does there exist a subset $S \subseteq V'$ of size at least k such that there is no edge in E' with both its endpoints in S?

2 X and Y are both in P

In this section, we consider the case that CCAV and CCDV for both X and Y are polynomial-time solvable.

Theorem 1. *CCAV and CCDV for Identity* THEN *1-Approval are polynomial-time solvable.*

Proof. In an Identity election, all candidates participating the election are winners and CCAV and CCDV for 1-Approval are polynomial-time solvable [2]. Identity THEN 1-Approval is in fact a 1-Approval election as a whole. Hence, CCAV and CCDV for Identity THEN 1-Approval are polynomial-time solvable. □

Corollary 1. *CCAV and CCDV for Identity* THEN *Identity are polynomial-time solvable.*

Theorem 2. *CCAV and CCDV for 1-Approval* THEN *1-Approval are NP-hard.*

Proof. First we consider CCAV. CCAV for 1-Approval is polynomial-time solvable [2]. We reduce from VERTEX COVER problem on 3-regular graphs.

Given a 3-regular graph $G = (V', E')$ where $|V'| = n$ and $|E'| = m$, we create an instance (C, V, W, p, k) for CCAV of 1-Approval THEN 1-Approval in the following way. Our election has the following candidates: p is the preferred candidate. For each edge $e'_i \in E'$, we have a candidate $e_i \in C$, and for each vertex $v'_i \in V'$, we have a candidate $v_i \in C$. Moreover, we have a dummy candidate d which will lose the initial round. We have the following registered votes (here "..." means that the remaining candidates are ordered arbitrarily):

- For every i, $1 \leq i \leq m$, four registered votes: $e_i \succ \ldots$
- Four registered votes: $p \succ \cdots$
- Two registered votes: $d \succ p \succ \cdots$
- For every $v'_i \in V$ such that e'_{i1}, e'_{i2} and e'_{i3} are the edges incident to v'_i, we have the following three registered votes: $v_i \succ e_{i1} \succ \ldots$, $v_i \succ e_{i2} \succ \ldots$ and $v_i \succ e_{i3} \succ \ldots$.

Now, we create a set W of unregistered votes by adding one vote $v_i \succ \ldots$ for every $1 \leq i \leq n$. Next, we show that G has a size-k vertex cover, if and only if p becomes the unique winner after adding k unregistered votes.

Note that every candidate in $C \setminus (\{d\} \cup \{v_i | v'_i \in V\})$ receives 4 approval votes from the registered votes in the initial round, while d receives 2 approval votes and every candidate in $\{v_i | v'_i \in V'\}$ receives 3 approval votes. If none of the unregistered votes is added, we have p and the candidates in $\{e_i | e'_i \in E'\}$ as winners in the initial round, and d and the candidates in $\{v'_i | v_i \in V'\}$ lose the initial round.

Let $S = \{v'_1, v'_2, \ldots, v'_k\}$ be a size-k vertex cover of G. For $1 \leq i \leq k$, let the ith unregistered vote $v_i \succ \ldots$ be added to the set of registered votes. We claim that by doing so, p becomes the winner in the overall election. It is obvious that after adding these votes, the winner set of the initial round is $C \setminus (\{d\} \cup \{v_i | v'_i \notin S\})$. Then, the set of votes should be restricted to these candidates. Since d loses the initial round, p gains in the second round two points from the votes $d \succ p \succ \ldots$. Note that after the addition of votes $v_i \succ \ldots$ for all v_i's with $v'_i \in S$, for each candidate in $\{e_i | e'_i \in E'\}$ there exists at least one candidate $v_i \in C$ such that v'_i is an endpoint of

e'_i and v_i is the winner in the initial round. Hence, each candidate in $\{e_i|e_i \in E'\}$ gets at most one additional vote in the second round after the candidates in $\{v_i|v'_i \notin S\}$ are omitted due to losing the initial round. Hence, in the second round, p gets totally 6 votes, each candidate in $\{e_i|e_i \in E'\}$ gets at most 5 votes and each candidate in $\{v_i|v'_i \in S\}$ gets 4 votes. Hence, p is the unique winner of the overall election.

For the reverse direction, recall that in the initial round, d receives only two votes and there are no unregistered votes which approve d. Hence, whatever k unregistered votes are added, d will lose the initial round and be omitted from the second round. By this, p gains in the second round two approval votes from the registered votes $d \succ p \succ \ldots$. Since we can only add k votes, at most k candidates in $\{v_i|v'_i \in V'\}$ will be among the winners of the initial round. Suppose there is a candidate v_j that is omitted from the second round of the election. Then, the candidates e_{j1}, e_{j2}, and e_{j3} which correspond to the edges incident to $v'_j \in V'$ gain one vote each in the second round from the votes approving v_j. Hence, the addition of k votes from W must assure that each candidate in $\{e_i|e'_i \in E'\}$ gains at most one vote by omitting the candidates in $\{v_i|v'_i \in V'\}$, whose scores remain the same after the addition of these k votes. This means that for each candidate in $\{e_i|e'_i \in E'\}$, at least one candidate v_i, whose corresponding vertex v'_i is one endpoint of e'_i, must be kept in the second round. Hence, the added votes must correspond to a vertex cover of G.

Next, we show the NP-hardness of CCDV. 1-Approval-CCDV is polynomial-time solvable [2]. We give a reduction from INDEPENDENT SET in 3-regular graphs to CCDV for 1-Approval THEN 1-Approval.

Given an instance $G = (V', E')$ where $|V'| = n$ and $|E'| = m$, we create an instance (C, V, p, k) of CCDV for 1-Approval THEN 1-Approval in the following way. Let p be the preferred candidate. For each $e'_i \in E'$, we create a candidate $e_i \in C$ and for each $v'_i \in V'$, create a candidate $v_i \in C$ and another dummy candidate x. Moreover, create another set of candidates $D = \{d, d_{i,j}, d_l\}$ for all $1 \leq i \leq m$, $1 \leq j \leq k^2 - 1$ and $1 \leq l \leq k^2$, which will lose in the initial round and never participate in the second round. Moreover, for every i we denote e_{m+i} as e_i. We have the following votes:

• A set P of $k + 4$ vote : $p \succ \ldots$
• A set Q containing three votes for each $v_i \in V$ such that e_{i1}, e_{i2} and e_{i3} are the edges incident to v_i :
$v_i \succ e_{i1} \succ e_{i1+1} \succ e_{i1+2} \succ \ldots \succ e_{i1+m} \succ \ldots \succ p$,
$v_i \succ e_{i2} \succ e_{i2+1} \succ e_{i2+2} \succ \ldots \succ e_{i2+m} \succ \ldots \succ p$ and
$v_i \succ e_{i3} \succ e_{i3+1} \succ e_{i3+2} \succ \ldots \succ e_{i3+m} \succ \ldots \succ p$.
• A set R containing one vote for each $1 \leq i \leq n$:
$v_i \succ x \succ e_1 \succ e_2 \succ \ldots \succ e_m \succ \ldots \succ p$
• A set X containing $k + 4$ votes of the following form:
$x \succ e_1 \succ e_2 \succ \ldots \succ e_m \succ \ldots \succ p$
• A set T containing $k + 4$ votes of the following form for each $e_i \in E$: $e_i \succ e_{i+1} \succ \ldots \succ e_{i+m} \succ \ldots \succ p$
• A set U containing k votes of the following form for each $v_i \in V$:
$v_i \succ p \succ \ldots$
• A set W of votes of the following form for each $1 \leq i \leq k$ and $1 \leq j \leq k^2 - 1$:
$d_{i,j} \succ e_i \succ e_{i+1} \succ e_{i+2} \succ \ldots \succ e_{i+m} \succ \ldots \succ p$
• A set Y with one vote for each $1 \leq l \leq k^2$:
$d_l \succ x \succ e_1 \succ e_2 \succ \ldots \succ e_m \ldots \succ p$
• A set Z with one vote: $d \succ p \succ \cdots$

With this construction and argument similar to above proof, we can show that CCDV for 1-approval THEN 1-Approval is NP-hard. We omit the detailed proof to the full version of the paper. □

Theorem 3. *CCAV for Veto* THEN *Veto is NP-hard, while CCDV for Veto* THEN *Veto is solvable in polynomial time.*

Proof. It is known that CCAV for Veto is polynomial-time solvable [2]. Now, we show that its runoff counterpart becomes hard. We reduce from the NP-complete VERTEX COVER on 3-regular graphs problem [15].

We reduce an instance $G = (V', E')$ to an instance (C, V, W, p, k) of CCAV for Veto THEN Veto in the following way: Let p be the preferred candidate. For each $e'_i \in E'$ create a candidate e_i and for each $v'_i \in V'$ create a candidate v_i. Create a dummy candidate l which will lose in the initial round and two other dummy candidates d and x. We have the following set of registered votes:
• For every i, $1 \leq i \leq n$, three registered votes:
$\ldots \succ p \succ e_{i1} \succ v_i$,
$\ldots \succ p \succ e_{i2} \succ v_i$, and
$\ldots \succ p \succ e_{i3} \succ v_i$,
where e'_{i1}, e'_{i2} and e'_{i3} are edges incident to v'_i in G.
• Three registered votes: $\cdots \succ p$
• Three registered votes: $\cdots \succ d$
• Three registered votes: $\cdots \succ x$
• Three registered votes for each $1 \leq i \leq m$: $\cdots \succ x \succ e_i$
• $n - k$ registered votes: $\cdots \succ d \succ l$
• $n - 1$ registered votes : $\cdots \succ p \succ l$
• $n - 1$ registered votes: $\cdots \succ e_i \succ l$
• n registered votes for each $1 \leq i \leq n$: $\cdots \succ x \succ v_i \succ l$

We also have a set W with one unregistered vote for each $1 \leq i \leq n$: $\cdots \succ d \succ v_i$. With this construction, and arguments similar to above theorem we can prove that CCAV for Veto THEN Veto is NP-hard. We omit the detailed proof to the final version of the paper.

Next, we consider CCDV. It is known that CCDV for Veto is polynomial-time solvable [2]. In contrast to CCAV, we prove that CCDV for Veto THEN Veto is polynomial-time solvable. Let p be the preferred candidate in the election. Without loss of generality, assume that if no vote is allowed to delete, p loses the election. Let W be the set of winners of the initial round if no vote is deleted. Let $\nu(c)$ denote the number of vetoes gained by c. For each candidate $c \in W$, if $\nu(p) - \nu(c) < k$, we can delete k votes which veto p. In this case p becomes the unique winner in the first round and the other candidates will be omitted. Otherwise, if $\nu(p) - \nu(w) = k$, the only way to prevent p from losing the initial round is to delete k votes that veto p. Now, in this case the only way to make p the unique winner in second round is by deleting the votes which veto p in such a way that each candidate $c \in W$ gains more vetoes than p in the second round from the remaining votes. The possibility of doing so can clearly be checked in polynomial time. Finally, if $\nu(p) - \nu(w) > k$, p will always lose the initial round and be omitted in the second round, a "no"-instance. □

3 One of X and Y is NP-hard

We move now to the case that for one of X and Y, we have NP-hard CCAV and CCDV.

Theorem 4. *CCAV and CCDV for Veto* THEN *4-Approval are NP-hard.*

Proof. We know that CCAV and CCDV for 4-Approval elections are NP-hard and for Veto are polynomial-time solvable [18]. The

NP-hardness reduction of CCAV and CCDV for 4-Approval can be modified to show the same result for Veto THEN 4-Approval. In the construction given in [18], we add a new candidate at the end of each vote. Now, since only these new candidates have veto score more than 0, all candidates in the original election are winners in the initial round. Then, in the second round, we have an instance of CCAV/CCDV for 4-Approval. □

With the same construction, we can prove that CCAV and CCDV for Veto THEN Y are NP-hard for many other elections Y with NP-hard CCAV and CCDV such as Copeland, Condorcet, Maximin, Borda etc.

Theorem 5. *CCAV and CCDV for 4-Approval* THEN *1-Approval are NP-hard.*

Proof. CCAV for 4-Approval is NP-hard, but for 1-Approval is polynomial-time solvable [18]. We give a reduction from an X3C-instance $(\mathcal{B}, \mathcal{S})$ to an instance (C, V, W, p, k) of CCAV for 4-Approval THEN 1-Approval which is similar to the NP-hardness reduction for CCAV for 4-Approval in [18].

Without loss of generality, we assume that $n \geq m$ and $3m$ mod 4 = 0. Recall that $n = |\mathcal{S}|$ and $3m = |\mathcal{B}|$. We construct the following instance (C, V, W, p, m) of CCAV for 4-Approval THEN 1-Approval. We set $C = \{p\} \cup \{b_i | B_i \in \mathcal{B}\}$. We assign registered votes to V such that in the initial 4-Approval election, p receives no approval and b_i receives $m - 1$ approvals for each $1 \leq i \leq 3m$. This is possible since $3m$ mod 4 = 0. Thus there are $3m + 1$ candidates and $\frac{3m(m-1)}{4}$ registered votes. For each $S_i \in \mathcal{S}$ with $S_i = \{B_{i1}, B_{i2}, B_{i3}\}$, we add an unregistered vote $v_i := p \succ b_{i1} \succ b_{i2} \succ b_{i3} \succ \ldots$ to W. We can show that $(\mathcal{B}, \mathcal{S})$ has an exact covering if and only if it is possible to make p the winner by adding m votes from W.

Next, we consider CCDV. CCDV is NP-hard for 4-approval [18]. We give a reduction from RX3C to CCDV for 4-Approval THEN 1-Approval. Let $(\mathcal{B}, \mathcal{S})$ be the input instance, where $\mathcal{B} = \{B_1, \ldots, B_{3m}\}$ and $\mathcal{S} = \{S_1, \ldots, S_n\}$ is a collection of 3-element subsets of $\mathcal{B}$.

Given this instance of RX3C, we construct the following instance (C, V, p, m) of CCDV for 4-Approval THEN 1-Approval. We set $C = \{p, p', p'', p''''\} \cup \{b_i | B_i \in \mathcal{B}\} \cup \{x_1, \ldots, x_m\}$. For each 3-set $S_i = \{B_{i1}, B_{i2}, B_{i3}\}$ we add a vote whose the first three preferred candidates are $\{b_{i1}, b_{i2}, b_{i3}\}$ and the fourth preferred candidate is x_i, while the remaining candidates are in arbitrary order. We create these votes in such a way that no candidate in $\{b_1, \ldots, b_{3m}\}$ appears in two of these votes as the first preferred candidate. Finally, we add two votes with the first four preferred candidates $p \succ p' \succ p'' \succ p'''$ while the remaining part of these votes is in the arbitrary order. □

Theorem 6. *There exist rules X and Y, such that CCAV and CCDV are polynomial-time solvable for X and X* THEN *Y but NP-hard for Y.*

Proof. We consider X being 1-Approval* election, where 1-Approval* is the special case of 1-Approval which breaks ties in favor of the preferred candidate, and Y being 4-Approval election. We know that CCAV and CCDV for 1-Approval are polynomial-time solvable [18] and the same algorithm can be applied for 1-Approval* as well. CCAV and CCDV for 4-Approval are NP-hard [18]. Observe that p must be the unique winner in the initial stage and all the remaining candidates are omitted in the second stage. Hence, polynomial-time algorithms for CCAV and CCDV for 1-Approval* can also be applied for the overall election. □

Theorem 7. *There exist rules X and Y, such that CCAV and CCDV are NP-hard for X but polynomial-time solvable for Y and X* THEN *Y.*

Proof. We consider X= *α-Condorcet*, which is the multi-winner voting rule that given a set of candidates, each assigned a label elects both the Condorcet winner and the candidate with the *lexicographically smallest label*. The *lexicographic order* on labels is the relation defined by $i < j$, if i comes (strictly) before j in the dictionary. Let Y be the election rule which elects the candidate with the lexicographically smallest label. Now, since CCAV and CCDV for Condorcet Voting are NP-hard [2], the same holds for *α-Condorcet* as well. With these two rules, X THEN Y elects a candidate with the lexicographically smallest label. Hence, CCAV and CCDV are polynomial-time solvable for Y and X THEN Y. □

Theorem 8. *There exist rules X and Y, such that CCAV and CCDV are NP-hard for X and X* THEN *Y but polynomial-time solvable for Y.*

Proof. We consider X= Condorcet, which always elects the Condorcet winner, and Y= Identity. CCAV and CCDV for Condorcet Voting are NP-hard [2] but polynomial-time solvable for Identity. It is easy to observe that the overall election is Condorcet and hence both CCAV and CCDV for X THEN Y are NP-hard. □

4 X and Y are both NP-hard

Finally, we study X THEN Y, where CCAV and CCDV for both X and Y are NP-hard.

Theorem 9. *CCAV and CCDV for r-Approval* THEN *Condorcet are NP-hard.*

Proof. CCAV for r-Approval election with $r \geq 4$ and Condorcet voting are NP-hard [18, 2]. We give a reduction from RX3C. Let $(\mathcal{B}, \mathcal{S})$ be an input instance of RX3C, where $\mathcal{B} = \{B_1, \ldots, B_{3m}\}$ and $\mathcal{S} = \{S_1, \ldots, S_{3m}\}$ is a collection of 3-element subsets of $\mathcal{B}$. Next, we construct the following instance (C, V, W, p, m) of CCAV for r-Approval THEN Condorcet. For an instance of RX3C, create an election with candidates p, b_i's for $i = 1, \ldots, 3m$ and a dummy candidate d. Let V consist of $m - 3$ voters, all with the preference $b_1 \succ \ldots \succ b_{3m} \succ p \succ d$. Let W contain one unregistered voter corresponding to each $S_j \in \mathcal{S}$ with preference $b_{j1} \succ b_{j2} \succ b_{j3} \succ p \succ B \setminus \{b_{j1} \cup b_{j2} \cup b_{j3}\} \succ d$, where the first three candidates correspond to the elements in S_j and the candidates between p and d are in arbitrary order. Now, in the above election we claim that p can become the unique winner of r-Approval THEN Condorcet where $r = 3m + 1$ if and only if there is a solution to RX3C of size $k = m$.

Similar to the proof of Theorem 9, we can modify the proof of the NP-hardness of CCDV for Condorcet in [12] to show the NP-hardness of CCDV for r-Approval THEN Condorcet where $r = 3m + 1$. □

We remark that CCDV for r-Approval THEN Y is NP-hard for many other voting rules Y with NP-hard CCDV such as Copeland, Borda, Maximin, etc.

Theorem 10. *CCAV and CCDV for 4-Approval* THEN *4-Approval are NP-hard .*

Proof. CCAV for 4-Approval election is NP-Hard [18]. We give a reduction from X3C to CCAV for 4-Approval THEN 4-Approval. Let

$(\mathcal{B}, \mathcal{S})$ be an input instance of X3C, where $\mathcal{B} = \{B_1, \ldots, B_{3m}\}$ and $\mathcal{S} = \{S_1, \ldots, S_n\}$ is a collection of 3-element subsets of $\mathcal{B}$. Without loss of generality we assume that $n \geq m$.

Next, we construct the following instance (C, V, W, p, m) for 4-Approval THEN 4-Approval. We set $C = \{p\} \cup \{b_i | B_i \in \mathcal{B}\} \cup \{d_1, d_2, \ldots, d_{m(m-1)}\}$. Thus, there are $m(m-1) + 3m + 2$ candidates. We construct the registered votes in V in such a way that in votes in V, p is always the fifth preferred candidates. Each candidate in $\{b_i | B_i \in \mathcal{B}\}$ occurs exactly $m-1$ times in the votes in V as one of the first three preferred candidates. This makes a total of $m(m-1)$ votes in V. The ith vote in V has d_i as its fourth preferred candidate. For each $S_i \in \mathcal{S}$ with $S_i = \{B_{i1}, B_{i2}, B_{i3}\}$ in the instance of X3C, we add an unregistered vote v_i to W whose first four preferred candidates are $p \succ b_{i1} \succ b_{i2} \succ b_{i3}$. The rest of the candidates in these votes are in arbitrary order.

Next, we consider CCDV. CCDV for 4-Approval election is NP-hard [18]. We give a reduction from a RX3C instance $(\mathcal{B}, \mathcal{S})$ to an instance $(E = (C, V), p, k)$ of CCDV for 4-Approval THEN 4-Approval, which is similar to the reduction given for Theorem 5. We set $C = \{p, p', p'', p''', p''''\} \cup \{b_i | B_i \in \mathcal{B}\} \cup \{x_1, \ldots, x_m\}$. For each 3-set $S_i = \{B_{i1}, B_{i2}, B_{i3}\}$, we add one vote whose first three preferred candidates are $\{b_{i1}, b_{i2}, b_{i3}\}$ and next two preferred candidates are $x_i \succ p$. Finally, we add 2 votes with the first four candidates being $p \succ p' \succ p'' \succ p'''$. This makes a total of $m + 2$ votes in V. The rest of the candidates in these votes are in arbitrary order. □

Theorem 11. *There exist rules X and Y, such that CCAV and CCDV are NP-hard for both X and Y but polynomial-time solvable for X* THEN *Y.*

Proof. We set X = α-Condorcet and Y=α-k-*Approval*, which is the multi-winner voting rule that elects both the winner of k-*Approval* election and the candidate with the lexicographically smallest label. Moreover, if there are ties, then Y breaks the ties in favor of the candidate with the lexicographically smallest label. CCAV and CCDV in both Condorcet [2] and k-Approval [18] elections are NP-hard, and the same holds for α-*Condorcet* and α-k-*Approval* as well. With these two rules, X THEN Y elects a candidate with the lexicographically smallest label. Hence, CCAV and CCDV in X THEN Y are polynomial-time solvable. □

5 Conclusions

This paper explored the complexity of constructive control by adding votes and deleting votes in two-stage elections. Complementing the results on complexity of manipulating two-stage elections [20, 14], we observed that there is no general relation between the control complexity of the rules applied in the stages and the control complexity of the whole two-stage elections. The current work can be considered as the first step towards understanding the computational complexity of controlling two-stage voting rules. The next step could be to study the computational complexity of other control types, for instance, constructive controlling by adding and deleting candidates and destructive controlling with the same operations, in two-stage scenario. Another interesting research direction would be to study the complexity of control problems in two-stage elections in the presence of weighted votes. One could also study the complexity of bribery for two-stage elections. Finally, we leave the question open whether there exists a natural, concrete voting rule X such that CCAV and CCDC for X are NP-hard but for X THEN X become polynomial-time solvable.

REFERENCES

[1] Parimal Kanti Bag, Hamid Sabourian, and Eyal Winter, 'Multi-stage voting, sequential elimination and condorcet consistency', *J. Economic Theory*, **144**(3), 1278–1299, (2009).

[2] John J. Bartholdi, Craig A. Tovey, and Michael A. Trick, 'How hard is it to control an election', in *Mathematical and Computer Modeling*, pp. 27–40, (1992).

[3] Felix Brandt, Markus Brill, Edith Hemaspaandra, and Lane A. Hemaspaandra, 'Bypassing combinatorial protections: Polynomial-time algorithms for single-peaked electorates', in *AAAI*, (2010).

[4] Vincent Conitzer and Tuomas Sandholm, 'Nonexistence of voting rules that are usually hard to manipulate', in *AAAI*, pp. 627–634, (2006).

[5] Jessica Davies, Nina Narodytska, and Toby Walsh, 'Eliminating the weakest link: Making manipulation intractable?', in *AAAI*, (2012).

[6] Edith Elkind and Helger Lipmaa, 'Hybrid voting protocols and hardness of manipulation', in *ISAAC*, pp. 206–215, (2005).

[7] Gábor Erdélyi, Markus Nowak, and Jörg Rothe, 'Sincere-strategy preference-based approval voting fully resists constructive control and broadly resists destructive control', *Math. Log. Q.*, **55**(4), 425–443, (2009).

[8] Gábor Erdélyi and Jörg Rothe, 'Control complexity in fallback voting', in *CATS*, pp. 39–48, (2010).

[9] Piotr Faliszewski, Edith Hemaspaandra, and Lane A. Hemaspaandra, 'Using complexity to protect elections', *Commun. ACM*, **53**(11), 74–82, (2010).

[10] Piotr Faliszewski, Edith Hemaspaandra, and Lane A. Hemaspaandra, 'Multimode control attacks on elections', *J. Artif. Intell. Res. (JAIR)*, **40**, 305–351, (2011).

[11] Piotr Faliszewski, Edith Hemaspaandra, and Lane A. Hemaspaandra, 'Weighted electoral control', in *AAMAS*, pp. 367–374, (2013).

[12] Piotr Faliszewski, Edith Hemaspaandra, Lane A. Hemaspaandra, and Jörg Rothe, 'Llull and copeland voting computationally resist bribery and constructive control', *J. Artif. Intell. Res. (JAIR)*, **35**, 275–341, (2009).

[13] Peter C. Fishburn, 'Condorcet social choice functions', *SIAM Journal on Applied Mathematics*, **33**(3), pp. 469–489, (1977).

[14] Zack Fitzsimmons, Edith Hemaspaandra, and Lane A. Hemaspaandra, 'X THEN X: Manipulation of same-system runoff elections', *CoRR*, **abs/1301.6118**, (2013).

[15] M. R. Garey, David S. Johnson, and Larry J. Stockmeyer, 'Some simplified np-complete problems', in *STOC*, pp. 47–63. ACM, (1974).

[16] Teofilo F. Gonzalez, 'Clustering to minimize the maximum intercluster distance', *Theor. Comput. Sci.*, **38**, 293–306, (1985).

[17] Edith Hemaspaandra, Lane A. Hemaspaandra, and Jörg Rothe, 'Anyone but him: The complexity of precluding an alternative', *Artif. Intell.*, **171**(5-6), 255–285, (2007).

[18] Andrew Lin, 'The complexity of manipulating k-approval elections', in *ICAART (2)*, pp. 212–218, (2011).

[19] Curtis Menton, 'Normalized range voting broadly resists control', *Theory Comput. Syst.*, **53**(4), 507–531, (2013).

[20] Nina Narodytska and Toby Walsh, 'Manipulating two stage voting rules', in *AAMAS*, pp. 423–430, (2013).

[21] David C. Parkes and Lirong Xia, 'A complexity-of-strategic-behavior comparison between schulze's rule and ranked pairs', in *AAAI*, (2012).