Anyone But Them: The Complexity Challenge for A Resolute Election Controller

Yongjie Yang
[1]School of Information Science and Engineering
Central South University, Changsha, China
[2]Chair of Economic Theory
Saarland University, Saarbrücken, Germany
yyongjiecs@gmail.com

Jianxin Wang
School of Information Science and Engineering
Central South University, Changsha, China
jxwang@csu.edu.cn

ABSTRACT

We study the voting problems where given is an election associated with a subset J of candidates, and the question is whether we can modify the election in a way so that none of the candidates in J wins the election. The modification operations include either adding some votes/candidates or deleting some votes/candidates. These problems are natural generalizations of destructive control problems where J is a singleton and capture many practical situations. We achieve a broad range of complexity results for a number of single-winner voting systems involving voting rules which are compositions of commonly used voting correspondences, such as Borda, Maximin and Copeland$^{\alpha}$, and three tie-breaking schemes, namely the fixed-order, random candidates and random votes. In particular, we achieve polynomial-time solvability results, NP-hardness results, fixed-parameter tractability results as well as XP results. In addition, we study other tie-breaking schemes and show that the complexity of the problems may depend on tie-breaking schemes.

Keywords

destructive control; voting systems; Borda; Maximin; Copeland; parameterized complexity

1. INTRODUCTION

Voting is a common method for preference aggregation and collective decision-making, and has significant applications in multiagent systems, political elections, web spam reduction, pattern recognition, etc. For instance, in multiagent systems, it is often necessary for a group of agents to make a collective decision by means of voting in order to reach a joint goal. In real-world applications, there exist many potential factors that may affect the result of voting. For instance, an external agent may add some new voters/candidates or delete some voters and candidates. These scenarios have been formulated as strategic voting problems and extensively studied in the literature [2, 3, 5, 7, 15, 23, 31, 34]. A prominent method to address such issues concerning strategic behavior is to use complexity as a barrier [3, 11, 22]. The key point is that if it is computationally hard for the external agent to figure out how to successfully change the result, he may refrain from attacking the voting.

Appears in: *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2017), S. Das, E. Durfee, K. Larson, M. Winikoff (eds.), May 8–12, 2017, São Paulo, Brazil.*

Typical goals of the external agent are to make some distinguished candidate p win the election or make p lose the election. A strategic voting problem with the former goal is referred to as a constructive problem and with the latter one a destructive problem. Constructive strategic voting problems have long been studied since the conducted work of Bartholdi, Tovey and Trick [2, 3], while the study of destructive counterpart was postponed till the works of Conitzer, Sandholm and Lang [6], and Hemaspaandra, Hemaspaandra and Rothe [15]. The motivation of the study of destructive strategic voting problems can be summarized as follows. First, the external agent may have a strong will to prevent someone from winning. Second, constructive strategic voting problems often turned out to be computationally harder than their destructive counterparts, see, e.g., [4, 9, 33]. As a consequence, the external agent who is a strong supporter of some candidate may find it is more efficient to indirectly support his favorite candidate by carrying out some strategic operations against some other candidates.

In this paper, we mainly study destructive strategic voting problems. However, in contrast to previous studies where the external agent desires to prevent only one distinguished candidate from winning the election, we consider the general case where there is a set J of distinguished candidates whom the external agent wants to make lose the election. This generalization captures more real-world situations. For instance, in voting with multi-issue domains, each candidate has a multi-issue structure and is uniquely identified by values that these issues take. In this case, the external agent may averse a specific value of an issue and hence wants to prevent all candidates taking this value from winning. Multi-issue domains have been extensively studied in the literature. In addition, such a generalization makes much sense concerning the second aspect of motivation discussed above. Moreover, for both theoretical and practical interests, the size of J provides a natural parameter for the study of the problems from the parameterized complexity point of view. For instance, in many real-world voting, there may be only a few competitive candidates and other candidates have small chance to win. If the resource is limited (e.g., lack of enough money), the supporter of a specific candidate might focus on strategies only against (other) competitive candidates.

We consider four modification operations potentially carried out by the external agent, namely adding votes, deleting votes, adding candidates and deleting candidates, leading to four problems denoted by RCAV, RCDV, RCAC and RCDC, respectively. In general, each problem asks whether the external agent can make all candidates in J lose the election by carrying out the corresponding modification operation. We refer to Preliminaries for the formal definitions of these problems. We investigate the (parameterized) complexity of these problems for numerous single-winner voting systems, involving voting rules which are compositions of the prevalent voting correspondences Borda, Copeland and Maximin and some tie-breaking schemes, including the fixed-order, random candidates and random votes. We achieve a broad range of complexity results including polynomial-time solvability results, NP-hardness results, fixed-parameter tractability results as well as XP results.

Recall that a *parameterized problem* is a subset $Q \subseteq \Sigma^* \times \mathbb{N}$ for some finite alphabet Σ, where the first component is the *main part* and the second component is the *parameter*. A parameterized problem is *fixed-parameter tractable (FPT)* if it is solvable in $O(f(k)|n|^{O(1)})$ time, and is in XP if it is solvable in $O(|n|^{f(k)})$ time, where k is the parameter, $|n|$ is the size of the main part and f is a computable function in k. A parameterized problem is *para-NP-hard* if it is NP-hard even for constant parameters.

We first show that, except for RCDC for Maximin, all problems are NP-hard. Then, we investigate how the size of J affects the parameterized complexity of the problems. On the one hand, we show that some of these problems are para-NP-hard. On the other hand, we show all the remaining problems become polynomial-time solvable if J contains a constant number of candidates. In the parameterized complexity language, we show that they are either fixed-parameter tractable, or fall in the class XP. In particular, for RCAC for Maximin we derive a single-exponential time algorithm with running time $O^*(2^k)$[1]. See Table 1 for a summary of our complexity results. Our results hold for all aforementioned tie-breaking schemes. However, we study an artificially made tie-breaking scheme and show that there are problems whose complexity differs with respect to this tie-breaking scheme and some of the aforementioned tie-breaking schemes (see Theorems 3, 9 and 10).

The main reason that we study Borda, Copeland and Maximin is that they are among the most significant voting correspondences that have been extensively studied in the literature. In particular, Borda is arguably the most classic positional voting correspondence, and Copeland and Maximin are well-studied Condorcet-consistent voting correspondences. Recall that the Condorcet winner is the candidate who is preferred to every other candidate by a majority of the voters. It is widely believed that Condorcet winner outperforms all other candidates in many senses. Unfortunately, the Condorcet winner does not always exist. A voting correspondence is Condorcet-consistent if it selects the Condorcet winner whenever it exists. We would like to point out that studying the same problems for other voting such as Bucklin and Approval is also an interesting topic.

1.1 Preliminaries

Voting Correspondence. A *voting* is specified by an ordered set $\mathcal{C}$ of *candidates*, a multiset $\Pi_\mathcal{V} = \{\succ_{v_1}, \succ_{v_2}, ..., \succ_{v_n}\}$ of *votes* cast by a corresponding set $\mathcal{V} = \{v_1, v_2, \dots, v_n\}$ of *voters* ($\succ_{v_i}$ is cast by v_i), and a *voting correspondence* φ which maps the *election* $\mathcal{E} = (\mathcal{C}, \Pi_\mathcal{V}, \mathcal{V})$ to a nonempty subset of candidates $\varphi(\mathcal{E})$, the *winners*. If there is only one winner, we call it the *unique winner*. We often discard $\mathcal{V}$ from the above notation for election $\mathcal{E}$ since $\Pi_\mathcal{V}$ is sufficient to determine the winners. Each vote $\succ_v \in \Pi_\mathcal{V}$ is defined as a linear order over the candidates, which is also referred to as the *preference* of the vote. If it is clear from the context, we drop v from $\succ_v$. We say a is *ranked above* b in a vote $\succ_v$ if $a \succ_v b$. Throughout this paper, we interchangeably use the terms "vote" and "voter".

For a vote $\succ$ over $\mathcal{C}$ and $C \subseteq \mathcal{C}$, $\succ^C$ is the vote $\succ$ restricted to C, i.e., for $c, c' \in C$, $c \succ c'$ implies $c \succ^C c'$. Moreover, $(C, \Pi_\mathcal{V}^C)$ is the election $(\mathcal{C}, \Pi_\mathcal{V})$ restricted to C, i.e., an election with candidate set C and vote set $\{\succ^C | \succ \in \Pi_\mathcal{V}\}$. For simplicity, we write $(C, \Pi_\mathcal{V})$ for $(C, \Pi_\mathcal{V}^C)$.

[1] $O^*()$ is the $O()$ notation with suppressed factors polynomial in the size of the input.

For two candidates c and c' in an election $\mathcal{E} = (\mathcal{C}, \Pi_\mathcal{V})$, $N_\mathcal{E}(c, c')$ is the number of votes preferring c to c' in $\Pi_\mathcal{V}$. We drop the index $\mathcal{E}$ when it is clear from context. If $N_\mathcal{E}(c, c') > N_\mathcal{E}(c', c)$, we say c *beats* c'; otherwise if $N_\mathcal{E}(c, c') = N_\mathcal{E}(c', c)$ we say c *ties* c' in $\mathcal{E}$. In this paper, we mainly study the following voting correspondences.

Borda: Every voter gives 0 points to her last-ranked candidate, 1 point to her second-last ranked candidate and so on. A candidate with the highest score is a winner.

Copeland$^\alpha$: For a candidate c, let $B(c)$ and $T(c)$ be the sets of candidates who are beaten by c and who tie with c, respectively. The Copeland$^\alpha$ score of c is $|B(c)| + \alpha \cdot |T(c)|$. Here, α is a rational number such that $0 \leq \alpha \leq 1$. A Copeland$^\alpha$ winner is a candidate with the highest score.

Maximin: For a candidaet c, the Maximin score of c is defined as $\min_{c' \in \mathcal{C} \setminus \{c\}} N(c, c')$. A Maximin winner is a candidate with the highest Maximin score.

Voting Rule. A *single-winner voting* is a voting with the voting correspondence being replaced with a *voting rule*. A *voting rule* $\psi = \mu \circ \varphi$ is the composition of a voting correspondence φ and a tie-breaking scheme μ which assigns to each $C \subseteq \mathcal{C}$ a probability distribution over C and selects exactly one winner according to the probability distribution. For simplicity, let $\mu(C, c)$ be the probability μ assigns to $c \in C$. Hence, $\sum_{c \in C} \mu(C, c) = 1$.

Tie-Breaking Scheme. The following tie-breaking schemes are widely studied in the literature. Let $(\mathcal{C}, \Pi_\mathcal{V})$ be an election and $C \subseteq \mathcal{C}$ a subset of candidates.

Fixed-Order μ^{FO}: There is a pre-defined order over $\mathcal{C}$, and for each $c \in C$, $\mu^{\text{FO}}(C, c) = 1$ if and only if c is ranked above every other candidate in C in the pre-defined order.

Random Candidates μ^{RC}: For each $c \in C$, $\mu^{\text{RC}}(C, c) = 1/|C|$.

Random Votes μ^{RV}: For each $c \in C$, let n_c be the number of votes that rank c in the top among all candidates in C. Then, $\mu^{\text{RV}}(C, c) = n_c / |\Pi_\mathcal{V}|$.

Problem Formulation. We first formulate a general problem.

Multimode Control

Input: An ordered set $\mathcal{C}$ of candidates, a multiset $\Pi_\mathcal{V}$ of votes over $\mathcal{C}$, a subset $\mathcal{D} \subseteq \mathcal{C}$, a nonempty subset $J \subseteq \mathcal{C} \setminus \mathcal{D}$, a submultiset $\Pi_\mathcal{U} \subseteq \Pi_\mathcal{V}$, positive integers $k_{\text{AV}} \leq |\Pi_\mathcal{U}|, k_{\text{DV}} \leq |\Pi_\mathcal{V} \setminus \Pi_\mathcal{U}|, k_{\text{AC}} \leq |\mathcal{D}|, k_{\text{DC}} \leq |\mathcal{C} \setminus (\mathcal{D} \cup \mathcal{J})|$, a real number $0 \leq \rho \leq 1$, a voting correspondence φ and a tie-breaking scheme μ.

Question: Are there $D \subseteq \mathcal{D}, C \subseteq \mathcal{C} \setminus (\mathcal{D} \cup J), \Pi_V \subseteq \Pi_\mathcal{V} \setminus \Pi_\mathcal{U}, \Pi_U \subseteq \Pi_\mathcal{U}$ such that $|D| \leq k_{\text{AC}}, |C| \leq k_{\text{DC}}, |\Pi_U| \leq k_{\text{AV}}, |\Pi_V| \leq k_{\text{DV}}$ and $\mu(\varphi(A, \Pi_\mathcal{F}), c) \leq \rho$ for every $c \in J \cap \varphi(A, \Pi_\mathcal{F})$, where $A = ((\mathcal{C} \setminus \mathcal{D}) \setminus C) \cup D$ and $\Pi_\mathcal{F} = ((\Pi_\mathcal{V} \setminus \Pi_\mathcal{U}) \setminus \Pi_V) \cup \Pi_U$?

In the definition, candidates in $\mathcal{D}$ and votes in $\Pi_\mathcal{U}$ are referred to as *unregistered candidates* and *unregistered votes*, respectively. Candidates not in $\mathcal{D}$ and votes not in $\Pi_\mathcal{U}$ are referred to as *registered candidates* and *registered votes*, respectively. Moreover, candidates in J are *distinguished candidates*. In general, the question asks, given the original election $(\mathcal{C} \setminus \mathcal{D}, \Pi_\mathcal{V} \setminus \Pi_\mathcal{U})$, whether an external agent can add at most k_{AV} unregistered votes and k_{AC} unregistered candidates, delete at most k_{DV} registered votes and k_{DC}

registered candidates, so that every candidate in J has a possibility less than ρ to win the election.

Resolute Control. Resolute control problems are special cases of the multimode control problem. Hence, we only describe how the inputs are restricted in each problem. First, in all resolute control problems we have $\rho = 0$. That is, the external agent has a "resolute" goal to prevent all distinguished candidates from having any chance to win. Clearly, NP-hardness for $\rho = 0$ implies the NP-hardness for ρ being a part of the input. Moreover, in each resolute control problem, the external agent may perform only one of the following four modification operations: adding votes, deleting votes, adding candidates and deleting candidates. Therefore, we have in total four problems—RCAV, RCDV, RCAC and RCDC. Here "RC" stand for "resolution control" and "AV|DV|AC|DC" stand for "adding votes|deleting votes|adding candidates|deleting candidates". Precisely, for each $X \in \{\text{AV, DV, AC, DC}\}$, in the input of the RCX problem, $k_Y = 0$ for every $Y \in \{\text{AV, DV, AC, DC}\} \setminus \{X\}$. Moreover, for $X \in \{\text{AV, DV, DC}\}$, $\mathcal{D} = \emptyset$, and for $X \in \{\text{AC, DC, DV}\}$, $\Pi_{\mathcal{U}} = \emptyset$.

Throughout this paper, $I = (\mathcal{C}, \Pi_{\mathcal{V}}, \mathcal{D} \subseteq \mathcal{C}, J \subseteq \mathcal{C} \setminus \mathcal{D}, \Pi_{\mathcal{U}} \subseteq \Pi_{\mathcal{V}}, k_{\text{AV}}, k_{\text{DV}}, k_{\text{AC}}, k_{\text{DC}}, \rho = 0, \varphi, \mu)$ is the given instance of the resolute control problem under consideration. We will not state this again in all proofs. In addition, RCX for $\mu \circ \varphi$ is RCX with voting correspondence and tie-breaking scheme φ and μ, respectively. For simplicity, when mentioning an instance of a problem in a proof, we ignore $\mu \circ \varphi$ and the components with values 0 or $\emptyset$. Moreover,

$$m = |\mathcal{C}|, m' = |\mathcal{D}|, n = |\Pi_{\mathcal{V}}|, n' = |\Pi_{\mathcal{U}}|, k = |J|.$$

1.2 Related Work

Our work is closely related to constructive/destructive control by adding/deleting votes/candidates extensively studied in the literature. The abbreviated names of these control problems are "CCAV, CCDV, CCAC, CCDC, DCAV, DCDV, DCAC, DCDC", where the first two characters "CC|DC" stand for "constructive control|destructive control" and the last two characters "AV|DV|AC|DC" stand for "adding votes|deleting votes|adding candidates|deleting candidates". In particular, for each $X \in \{\text{AV}, \text{DV}, \text{AC}, \text{DC}\}$, CCX (resp. DCX) has the same input as that of RCX but with $J = \{p\}$, and the question is whether p can win (resp. lose) the election by carrying out the corresponding modification operations (see [3, 13, 15] for further details). Faliszewski, Hemaspaandra and Hemaspaandra [12] studied a generalization of the above control problems where the external agent launches a multimode attack. In particular, in the generalization, all operations imposed in the above control problems, as well as operations of bribing votes are allowed. It is important to point out that control problems are mainly concerned with voting correspondences instead of voting rules. Hence, two different models, so-called the *unique-winner* and *nonunique-winner* models, used to specify the exact meaning of winning an election are often simultaneously studied for each of the above problems in the literature. In particular, in the unique-winner model winning an election means to be the unique winner, and in the nonunique-winner model, winning an election means to be one of the winners (can be also the unique winner). The terms "UNI" and "NON" are appended to the names of the problems to specify the models of the problems. For instance, CCAV-UNI is the unique-winner model of the constructive control by adding votes problem, and CCAV-NON is the nonunique-winner model of the same problem. In discussions concerning control problems in this paper, results for CCX and DCX for every $X \in \{\text{AV}, \text{DV}, \text{AC}, \text{DC}\}$ hold for both the unique-winner model and the nonunique-winner model.

In general, the relation between control problems and resolute control problems are as follows. For $X \in \{\text{AV, DV}\}$ and many voting rules, by setting $J = (\mathcal{C} \setminus \mathcal{D}) \setminus \{p\}$, we can reduce CCX-UNI to RCX, where p is the distinguished candidate in the CCX-UNI instance and J is the set of distinguished candidates in the RCX instance. Notice that this does not apply to the case $X \in \{\text{AC, DC}\}$. The reason is that when $X = \text{AC}$, someone in $\mathcal{D}$ instead of p can also prevent all candidates in J from winning, and when $X = \text{DC}$ we are not allowed to delete candidates in J in RCX. Notice that tie-breaking schemes have impact on this relation. We will discuss this issue in detail in the next section. On the other hand, for $X \in \{\text{AV, DV, AC, DC}\}$, DCX-NON is a special case of RCX.

Since we focus on single-winner voting, it does not make sense to study constructive counterparts of the resolute control problems, i.e., to make all candidates in J win the election (if $|J| > 1$, the given instance must be a No-instance; otherwise, we have the constructive control problem). However, one may wonder the problems where the goal of the external agent is to make at least one candidate in J win the election. In fact, these are special cases of constructive control problems for multi-winner voting studied in [21, 22, 25, 26]. In this setting, the external agent has a utility to each candidate, and the goal of the external agent is to select exactly k' winners with a total utility greater than a given number R by adding/deleting votes/candidates. So, if the external agent has utility R to all candidates in J and has utility 0 to all the other candidates, we have the problems discussed above.

Our study is also related to the recent work by Erdélyi, Reger and Yang [10] where the destructive control problems in the setting of group identification were investigated. In group identification, voters and candidates coincide. Each voter either qualifies or disqualifies each candidate and the goal of group identification is to select a subset of individuals, which are referred to as socially qualified individuals. They studied destructive control problems where the goal of the external agent is to prevent a given subset of individuals from being socially qualified. Nevertheless, they are mainly concerned with the complexity of the problems, rather than the parameterized complexity of the problems with respect to the size of the given subset of individuals.

2. UNBOUNDED NUMBER OF DISTINGUISHED CANDIDATES

In this section, we investigate the complexity of resolute control problems when $|J|$ is not considered as a parameter. Let $X \in \{\text{AV}, \text{DV}\}$. We shall establish a connection between RCX and CCX-UNI in order to show the NP-hardness of RCX in general.

Suppose that a candidate p ties with several other candidates. Let μ be a tie-breaking scheme which breaks ties in a way so that a positive probability of p to win implies a positive probability of at least one of other tied candidates to win. Then to make p win with respect to $\mu \circ \varphi$ where φ is a voting correspondence means to make all except p lose the election with respect to φ. Due to this, we can reduce CCX-UNI to RCX by defining J as the set of all registered candidates except the distinguished candidate in the instance of CCX-UNI. As a consequence, the NP-hardness of CCX-UNI implies the NP-hardness of RCX.

Clearly, μ^{RC} satisfies the condition discussed above. For μ^{FO}, if p is ranked last in the pre-defined order, then μ^{FO} satisfies the condition. On the other hand, μ^{RV} does not satisfy the condition. Consider the situation where p is ranked above every other tied candidate in every vote. Then none of the tied candidates except p has a positive chance to win. Nevertheless, for many voting, two candidates c, c' can tie only when there are both votes ranking c

<table>
<tr><th rowspan="2"></th><th colspan="2">Maximin</th><th colspan="2">Copeland$^\alpha, 0 \le \alpha \le 1$</th><th colspan="2">Borda</th></tr>
<tr><th>k</th><th>general</th><th>k</th><th>general</th><th>k</th><th>general</th></tr>
<tr><td>RCAV
RCDV</td><td>pa-NP-h (Theorem 4)</td><td>NP-h (Theorem 1)</td><td>pa-NP-h (Theorem 4)</td><td>NP-h (Theorem 1)</td><td>XP (Theorem 7)</td><td>NP-h (Theorem 1)</td></tr>
<tr><td>RCAC</td><td>FPT (Theorem 5)
$O^*(2^k)$</td><td>NP-h (Theorem 2)</td><td rowspan="2">FPT (ILP-based)
(Theorem 6)</td><td rowspan="2">NP-h (Theorem 2)</td><td rowspan="2">XP (Theorem 8)</td><td rowspan="2">NP-h (Theorem 2)</td></tr>
<tr><td>RCDC</td><td colspan="2">P (Theorem 3)</td></tr>
</table>

Table 1: A summary of our results on the complexity of resolute control problems. All results hold for the 3 tie-breaking schemes described in Preliminaries. In the table, "P" stands for "polynomial-time solvable", "NP-h" for "NP-hard" and "pa-NP-h" for "para-NP-hard". Moreover, k is the number of distinguished candidates. Results in columns indicated by "general" means that we do not consider k as a parameter. All XP results are based on dynamic programming algorithms.

above c', and votes ranking c' above c. A voting correspondence φ is *Pareto optimal* if when all votes rank a candidate a above another candidate b, then b cannot be a winner with respect to φ.

LEMMA 1. *Let φ be a voting correspondence. For each $X \in \{AV, DV\}$, CCX-UNI for φ is polynomial-time reducible to RCX for $\mu^Y \circ \varphi$, where $Y \in \{FO, RC\}$, and for $\mu^{RV} \circ \varphi$ where φ is Pareto optimal.*

PROOF. We only show the reduction from CCAV-UNI to RCAV. Let $\mathcal{E} = (\mathcal{C}, \Pi_\mathcal{V}, p \in \mathcal{C}, \Pi_\mathcal{U}, k_{\text{AV}})$ be an instance of CCAV-UNI for φ, where p is the distinguished candidate. Consider first μ^{FO} and μ^{RC}. Clearly, $I_\mathcal{E} = (\mathcal{C}, \Pi_\mathcal{V}, J = \mathcal{C} \setminus \{p\}, \Pi_\mathcal{U}, k_{\text{AV}})$ is an instance of RCAV for $\mu^{\text{RC}} \circ \varphi$. The instance of RCAV for $\mu^{\text{FO}} \circ \varphi$ is $I_\mathcal{E}$ together with a pre-defined order where p is ranked last. The construction clearly takes polynomial time. The correctness of the reduction is easy to see. Consider now μ^{RV}. Clearly, $I_\mathcal{E}$ is an instance of RCAV for $\mu^{\text{RV}} \circ \varphi$. If $\mathcal{E}$ is a Yes-instance, so is $I_\mathcal{E}$. It remains to prove the opposite direction. Assume that we made all candidates in J have no positive probability to win with respect to $\mu^{\text{RV}} \circ \varphi$, by adding at most k_{AV} votes in $\Pi_\mathcal{U}$. Let A be the set of winners with respect to φ in the final election. If $p \notin A$, then $J \cap A \neq \emptyset$ and someone in J has a positive probability to win the final election with respect to $\mu^{\text{RV}} \circ \varphi$; a contradiction. So, assume that $p \in A$. If $A \cap J \neq \emptyset$, then since φ is Pareto optimal, not every vote ranks p above everyone in J in the final election. As a result, someone in J has a positive possibility to win the final election, a contradiction. In summary, p is the unique winner in the final election with respect to φ. □

It is known that CCAV and CCDV for Borda, Maximin and Copeland$^\alpha$ are NP-hard [3, 12, 27]. Then, due to Lemma 1 and the fact that Borda, Maximin and Copeland$^\alpha$ are Pareto optimal [28], we have the following theorem.

THEOREM 1. *For $Y \in \{FO, RC, RV\}$ and $\varphi \in \{$Borda, Maximin, Copeland$^\alpha\}$, RCAV and RCDV for voting rules $\mu^Y \circ \varphi$ are NP-hard.*

While Lemma 1 does not apply to control by adding/deleting candidates, we are able to show the NP-hardness of RCAC for Borda, Maximin and Copeland$^\alpha$, and the NP-hardness of RCDC for Borda and Copeland$^\alpha$.

THEOREM 2. *Let $Y \in \{FO, RC, RV\}$. Then, RCAC for $\mu^Y \circ \varphi$ where $\varphi \in \{$Borda, Maximin, Copeland$^\alpha\}$, and RCDC for $\mu^Y \circ \varphi$ where $\varphi \in \{$Borda, Copeland$^\alpha\}$ are NP-hard.*

Faliszewski, Hemaspaandra and Hemaspaandra [12] proved that both CCDC and DCDC for Maximin are polynomial-time solvable. We extend these results to resolute control problems.

THEOREM 3. *RCDC for $\mu^Y \circ$Maximin is polynomial-time solvable, for every $Y \in \{FO, RC, RV\}$.*

PROOF. We develop a polynomial-time algorithm as follows. The algorithm splits I into $m-k$ subinstances, each of which takes I together with a candidate $q \in \mathcal{C} \setminus J$ as the input, and asks if we can delete at most k_{DC} candidates in $\mathcal{C} \setminus (J \cup \{q\})$ so that the Maximin score of q is greater than that of everyone in J. Clearly, I is a Yes-instance if and only if at least one of the subinstances is a Yes-instance. Moreover, if each subinstance is solvable in $O(f(m,n))$ time, then I is solvable in $O(m \cdot f(m,n))$ time. Let $I' = (I, q)$ be a subinstance as discussed above. We solve I' as follows. Observe that deleting a candidate never decreases the scores of other candidates. Therefore, if there is a candidate $c \in \mathcal{C} \setminus (J \cup \{q\})$ such that $N(q,c)$ is less than the Maximin score of some $c' \in J$ at the moment, then c must be deleted. Due to this observation, to solve I', we need only to first order the candidates $c \in \mathcal{C} \setminus (J \cup \{q\})$ according to a non-decreasing order of $N(q,c)$. Then, we delete the first $\min\{k_{\text{DC}}, m-|J|-1\}$ candidates in this order one by one: if q has Maximin score greater than that of every candidate in J after deleting a candidate, we immediately conclude that I' is a Yes-instance. If no Yes-instance is concluded after deleting these candidates, I' is a No-instance. The algorithm clearly terminates in polynomial time. □

3. BOUNDED NUMBER OF DISTINGUISHED CANDIDATES

In this section, we show how the number of distinguished candidates, i.e., $k = |J|$, affects the complexity of resolute control problems. In particular, for each problem studied in this paper and constant k, we either show it is NP-hard, i.e., the problem is para-NP-hard with respect to k, or develop a polynomial-time algorithm, i.e., the probelm is FPT or in XP with respect to k. Most of our FPT- and XP-algorithms begin with splitting the given instance I into polynomially many subinstances (no more than m), each of which takes as input I together with a candidate $q \in \mathcal{C} \setminus J$, and asks whether q can prevent all candidates in J from winning the election by carrying out the corresponding modification operations imposed in the problem (analogous to the algorithm in the proof of Theorem 3). Clearly, I is a Yes-instance if and only if at least one of the subinstances is a Yes-instance. Moreover, if each subinstance can be solved in FPT-time (XP-time), then I can be solved in the same time complexity with respect to the notation $O^*()$. For simplicity, we will not describe this step in all algorithms shown below (except the one in the proof of Theorem 5, in which we split I in a little different way) and will mainly focus on algorithms to solve each subinstance. Hereinafter, $I' = (I, q \in \mathcal{C} \setminus J)$ is a subinstance of I as discussed above.

3.1 Copeland and Maximin

We first investigate Copeland and Maximin both of which are Condorcet-consistent. It is known that DCAV-NON and DCDV-NON for Maximin and Copland$^\alpha$ are NP-hard [12, 13]. As DCAV-NON and DCDV-NON are special cases of RCAV and RCDV respectively with J being a singleton, it follows that RCAV and RCDV for $\mu^Y \circ Copeland^\alpha$ and $\mu^Y \circ Maximin$, where $Y \in \{\text{FO, RC, RV}\}$, are NP-hard even if $k = 1$. From the parameterized complexity perspective, they are para-NP-hard.

THEOREM 4 ([12, 13]). *RCAV and RCDV for $\mu^Y \circ Copeland^\alpha$ and $\mu^Y \circ Maximin$, where $Y \in \{FO, RC, RV\}$, are para-NP-hard with respect to k.*

Now we turn our attention to RCAC. We have shown that RCAC for Maximin in general (unbounded k) is NP-hard (Theorem 2). We prove now that RCAC for Maximin is FPT with respect to k. In particular, we develop a dynamic programming algorithm with running time $O^*(2^k)$ for the problem.

THEOREM 5. *RCAC for $\mu^Y \circ$ Maximin is FPT with respect to k, for every $Y \in \{FO, RC, RV\}$.*

PROOF. Let $I = (\mathcal{C}, \Pi_\mathcal{V}, \mathcal{D} \subseteq \mathcal{C}, J \subseteq \mathcal{C} \setminus \mathcal{D}, k_{\text{AC}})$ be a given instance. We give an algorithm as follows. The algorithm guesses a candidate $q \in \mathcal{C} \setminus J$, the final score sc of q and a candidate $q' \in \mathcal{C} \setminus \{q\}$ such that $N(q, q') = sc$. Assume that $\{q, q'\} \subseteq \mathcal{C} \setminus \mathcal{D}$, since otherwise, we can remove $\{q, q'\} \cap \mathcal{D}$ from $\mathcal{D}$ and decrease k_{AC} by $|\mathcal{D} \cap \{q, q'\}|$ (If $k_{\text{AC}} < 0$ after doing so, we discard the guess and proceed to the next one). This leads to at most $(m-k) \cdot (m-1) \cdot n$ subinstances, each of which takes I together with the guessed candidates q, q' and the guessed score sc as the input, and asks if we can add at most k_{AC} candidates in $\mathcal{D}$ so that the Maximin score of q is sc and the Maximin score of every candidate in J is less than sc. As Maximin is Pareto optimal, for every $Y \in \{\text{FO}, \text{RC}, \text{RV}\}$, I is a Yes-instance if and only if at least one of the subinstances is a Yes-instance. Let $I' = (I, q, q', sc)$ be a subinstance. Clearly, if there is a $c \in \mathcal{C} \setminus \mathcal{D}$ such that $N(q, c) < sc$, I' is a No-instance. Assume now that $N(q, c) \geq sc$ for every $c \in \mathcal{C} \setminus \mathcal{D}$. Let $(c_1, c_2, ..., c_t)$ be an arbitrary but fixed order of all candidates $c \in \mathcal{D}$ such that $N(q, c) \geq sc$, where $t \leq m'$. Let $J' \subseteq J$ be the set of candidates $c \in J$ such that there is a $c' \in \mathcal{C} \setminus \mathcal{D}$ such that $N(c, c') < sc$. Clearly, all candidates in J' have score less than sc no matter which candidates in $\{c_1, c_2, ..., c_t\}$ are added. So, we need only to focus on candidates in $J \setminus J'$. We maintain a boolean table $\text{DT}(i, R, B)$ where $1 \leq i \leq t, 1 \leq R \leq k_{\text{AC}}$, $B \subseteq J \setminus J'$ and $B \neq \emptyset$. The value of $\text{DT}(i, R, B)$ indicates whether we can add R candidates from $\{c_1, c_2, ..., c_i\}$ to make all candidates in B have scores less than sc. Due to the definition of DT, I' is a Yes-instance if and only if $\text{DT}(i, R, J \setminus J') = 1$ for some $1 \leq i \leq t$ and $0 \leq R \leq k_{\text{AC}}$. To calculate the entries of the table, for every nonempty $B \subseteq J \setminus J'$ we initialize $\text{DT}(1, 1, B) = 1$ if $N(c, c_1) < sc$ for every $c \in B$; and $\text{DT}(1, 1, B) = 0$ otherwise. In addition, $\text{DT}(1, 0, B) = 0$. Moreover, $\text{DT}(i, R, B) = 0$ if $R > i$. We use the following relation to update the table: $\text{DT}(i, R, B) = 1$ if and only if $\text{DT}(i-1, R, B) = 1$ or $\text{DT}(i-1, R-1, B \setminus M(c_i)) = 1$, where $M(c_i) = \{c \in J \setminus J' \mid N(c, c_i) < sc\}$. As we have $O^*(2^k)$ entries, the running time of the algorithm to solve I' is $O^*(2^k)$. As we have polynomially many subinstances, the running time of the algorithm to solve I is bounded by $O^*(2^k)$ as well. □

We have shown that RCAC and RCDC for Copeland$^\alpha$ are NP-hard in general. Now, we prove that when the number of distinguished candidates is considered as a parameter, both problems are FPT. In particular, we give an integer-linear programming formulation (ILP) with bounded number of variables for the instance we desire to solve. It is known that ILPs with z variables can be solved in FPT-time with respect to z [17].

THEOREM 6. *RCAC and RCDC for $\mu^Y \circ Copeland^\alpha$, $0 \leq \alpha \leq 1$ are FPT with respect to k, for every $Y \in \{FO, RC, RV\}$.*

PROOF. As discussed previously, we only show how to solve each subinstance $I' = (I, q)$ in FPT-time.

RCAC. If $q \in \mathcal{D}$, then we remove q from $\mathcal{D}$ and decrease k_{AC} by one. If $k_{\text{AC}} < 0$ after doing so, we immediately conclude that I' is a No-instance. Assume now that $q \in \mathcal{C} \setminus \mathcal{D}$. Observe that after adding a candidate $d \in \mathcal{D}$, the score of each candidate in $J \cup \{q\}$ either increases by one, or by α, or remains unchanged. Let $(c_1, c_2, ..., c_k)$ be an arbitrary but fixed order of J. For each candidate $d \in \mathcal{D}$, define $\vec{d}$ as a $(k+1)$-dimensional vector such that for each $1 \leq i \leq k$

$$\vec{d}[i] = \begin{cases} 1, & N(c_i, d) > N(d, c_i) \\ \alpha, & N(c_i, d) = N(d, c_i) \\ 0, & N(c_i, d) < N(d, c_i) \end{cases}$$

and

$$\vec{d}[k+1] = \begin{cases} 1, & N(q, d) > N(d, q) \\ \alpha, & N(q, d) = N(d, q) \\ 0, & N(q, d) < N(d, q) \end{cases}$$

Now we partition $\mathcal{D}$ into at most 3^{k+1} subsets. Precisely, two candidates $d, e \in \mathcal{D}$ are in the same subset if and only if $\vec{d} = \vec{e}$. Let $\mathcal{D}_{\vec{v}}$ denote the subset in the partition such that for each $d \in \mathcal{D}_{\vec{v}}$, it holds that $\vec{d} = \vec{v}$, where $\vec{v}$ is a $(k+1)$-dimensional vector whose components are from $\{1, \alpha, 0\}$. Now we give an ILP formulation with no more than 3^{k+1} variables for I'. For each subset $\mathcal{D}_{\vec{v}}$, we have a variable $x_{\vec{v}}$, indicating the number of candidates in $\mathcal{D}_{\vec{v}}$ to be added. The restrictions are as follows. Let $\mathcal{F} = \{\vec{d} \mid d \in \mathcal{D}\}$. Clearly, $|\mathcal{F}| \leq 3^{k+1}$.

(1) $0 \leq x_{\vec{v}} \leq |\mathcal{D}_{\vec{v}}|$ for every $\vec{v} \in \mathcal{F}$.

(2) Since we can add at most k_{AC} candidates, we have that

$$\sum_{\vec{v} \in \mathcal{F}} x_{\vec{v}} \leq k_{\text{AC}}.$$

For each $c \in \mathcal{C} \setminus \mathcal{D}$, let $sc(c)$ be the Copeland$^\alpha$ score of c in $(\mathcal{C} \setminus \mathcal{D}, \Pi_\mathcal{V})$. Notice that if $Y \in \{\text{RC, RV}\}$, then for q to prevent all candidates in J from winning, q has to have a Copeland$^\alpha$ score higher than that of every candidate in J in the final election. As a result, for each candidate $c_i \in J$, $1 \leq i \leq k$, it has to be that

$$sc(q) + \sum_{\vec{v} \in \mathcal{F}} \vec{v}[k+1] \cdot x_{\vec{v}} > sc(c_i) + \sum_{\vec{v} \in \mathcal{F}} \vec{v}[i] \cdot x_{\vec{v}}.$$

On the other hand, if $Y = \text{FO}$, then for q to prevent all candidates in J from winning, q has to have a Copeland$^\alpha$ score higher than that of every $c_i \in J$ such that $c_i \rhd q$, and have a score no less than that of every $c_i \in J$ such that $q \rhd c_i$, where $\rhd$ is the pre-defined tie-breaking order. Hence, for every $c_i \in J$ such that $c_i \rhd q$, it has to be that

$$sc(q) + \sum_{\vec{v} \in \mathcal{F}} \vec{v}[k+1] \cdot x_{\vec{v}} > sc(c_i) + \sum_{\vec{v} \in \mathcal{F}} \vec{v}[i] \cdot x_{\vec{v}}.$$

Moreover, for every $c_i \in J$ such that $q \rhd c_i$, it has to be that

$$sc(q) + \sum_{\vec{v} \in \mathcal{F}} \vec{v}[k+1] \cdot x_{\vec{v}} \geq sc(c_i) + \sum_{\vec{v} \in \mathcal{F}} \vec{v}[i] \cdot x_{\vec{v}}.$$

RCDC. The algorithm is similar to the above one for RCAC. Let $(c_1, c_2, ..., c_k)$ be an arbitrary but fixed order of J. Observe that after deleting a candidate $d \in \mathcal{C} \setminus (J \cup \{q\})$, the score of each candidate in $J \cup \{q\}$ either decreases by one, or by α, or remains unchanged. For each candidate $d \in \mathcal{C} \setminus (J \cup \{q\})$, let $\vec{d}$ be a $(k+1)$-dimensional vector defined in the same way as in the above algorithm for RCAC.

Now we partition $\mathcal{C} \setminus (J \cup \{q\})$ into at most 3^{k+1} subsets. Precisely, two candidates $d, e \in \mathcal{C} \setminus (J \cup \{q\})$ are in the same subset if and only if $\vec{d} = \vec{e}$. Let $\mathcal{C}_{\vec{v}}$ denote the subset in the partition such that for each $d \in \mathcal{C}_{\vec{v}}$, it holds that $\vec{d} = \vec{v}$, where $\vec{v}$ is a $(k+1)$-dimensional vector whose components are from $\{1, \alpha, 0\}$. Now we give an ILP formulation with no more than 3^{k+1} variables for I'. For each subset $\mathcal{C}_{\vec{v}}$, we have a variable $x_{\vec{v}}$, indicating the number of candidates in $\mathcal{C}_{\vec{v}}$ to be deleted from the election. The restrictions are as follows. Let $\mathcal{F} = \{\vec{d} \mid d \in \mathcal{C} \setminus (J \cup \{q\})\}$.

(1) For each variable $x_{\vec{v}}$, we have $0 \le x_{\vec{v}} \le |\mathcal{C}_{\vec{v}}|$.

(2) Since we can delete at most k_{DC} candidates, we have that

$$\sum_{\vec{v} \in \mathcal{F}} x_{\vec{v}} \le k_{\text{DC}}.$$

For each $c \in \mathcal{C}$, let $sc(c)$ be the Copeland$^\alpha$ score of c in the election $(\mathcal{C}, \Pi_{\mathcal{V}})$. Notice that if $Y \in \{\text{RC}, \text{RV}\}$, then for q to prevent all candidates in J from winning, q has to have a Copeland$^\alpha$ score higher than that of every candidate in J in the final election. As a result, for each candidate $c_i \in J$, it has to be that

$$sc(q) - \sum_{\vec{v} \in \mathcal{F}} \vec{v}[k+1] \cdot x_{\vec{v}} > sc(c_i) - \sum_{\vec{v} \in \mathcal{F}} \vec{v}[i] \cdot x_{\vec{v}}.$$

Analogous to the proof for RCAC, if $Y = \text{FO}$, then for every $c_i \in J$ such that $c_i \rhd q$, it has to be that

$$sc(q) - \sum_{\vec{v} \in \mathcal{F}} \vec{v}[k+1] \cdot x_{\vec{v}} > sc(c_i) - \sum_{\vec{v} \in \mathcal{F}} \vec{v}[i] \cdot x_{\vec{v}},$$

and for every $c_i \in J$ such that $q \rhd c_i$, it has to be that

$$sc(q) - \sum_{\vec{v} \in \mathcal{F}} \vec{v}[k+1] \cdot x_{\vec{v}} \ge sc(c_i) - \sum_{\vec{v} \in \mathcal{F}} \vec{v}[i] \cdot x_{\vec{v}}.$$

Due to the result of Lenstra [17], both ILPs for RCAC and RCDC constructed above are solvable in FPT time with respect to k. Hence, the problems stated in the theorem are FPT with respect to k. □

3.2 Borda

Consider now the non-Condorcet-consistent voting correspondence Borda. We have proved that RCAV and RCDV for $\mu^Y \circ Borda$, $Y \in \{\text{FO, RC, RV}\}$, are NP-hard if k is unbounded (Theorem 1). We prove now that if k is a constant, the same problems are polynomial-time solvable, based on dynamic programming algorithms. Precisely, from the parameterized complexity point of view, RCAV and RCDV for $\mu^Y \circ Borda$, $Y \in \{\text{FO, RC, RV}\}$, are in XP with respect to k. Recall that DCAV and DCDV for Borda are polynomial-time solvable [27]. Hence, our results generalize the results in [27].

THEOREM 7. *RCAV and RCDV for $\mu^Y \circ Borda$ are in XP with respect to k, for every $Y \in \{FO, RC, RV\}$.*

PROOF. As discussed previously, we show only how to solve each subinstance I' in XP-time. In particular, we resort to dynamic programming algorithms to solve I'. Let $(c_1, c_2, ..., c_k)$ be a fixed order of candidates in J. We first give the definitions of the tables and show how to calculate the entries as follows. For $c \in \mathcal{C}$ and $\Pi_{\mathcal{T}} \subseteq \Pi_{\mathcal{V}}$, let $scg(c, \Pi_{\mathcal{T}})$ be the score of c obtained from the votes in $\Pi_{\mathcal{T}}$ minus the score of q obtained from the votes in $\Pi_{\mathcal{T}}$. For each $c_i \in J$, let $\text{SC}(c_i) = scg(c_i, \Pi_{\mathcal{V}} \setminus \Pi_{\mathcal{U}})$. Recall that in RCDV we have $\Pi_{\mathcal{U}} = \emptyset$. For each $\succ_v \in \Pi_{\mathcal{V}}$, let $\overrightarrow{\succ_v} = \langle sc_1, ..., sc_k \rangle$ be the integer vector where $sc_i = scg(c_i, \{\succ_v\})$.

RCAV. Let $(\succ_{v(1)}, ..., \succ_{v(n')})$ be a fixed order of the unregistered votes. We maintain a table $\text{DT}(x, y, s_1, s_2, ..., s_k)$, where $1 \le x \le n$, $0 \le y \le k_{\text{AV}}$ and $|s_i| \le (m-1) \cdot (n - n' + k_{\text{AV}})$ for every $1 \le i \le k$. We define $\text{DT}(x, y, s_1, s_2, ..., s_k) = 1$ if and only if there exists $\Pi_V \subseteq \{\succ_{v(1)}, ..., \succ_{v(x)}\}$ such that $|\Pi_V| = y$ and for every $c_i \in J$ it holds that $\text{SC}(c_i) + scg(c_i, \Pi_V) = s_i$. Initially, we set $\text{DT}(1, 0, s_1, ..., s_k) = 1$ if and only if $\text{SC}(c_i) = s_i$ for every $1 \le i \le k$. In addition, $\text{DT}(1, 1, s_1, ..., s_k) = 1$ if and only if $\text{SC}(c_i) + \overrightarrow{\succ_{v(1)}}[i] = s_i$ for every $1 \le i \le k$. Moreover, for $x < y$ we set $\text{DT}(x, y, s_1, ..., s_k) = 0$. We use the following relation to update the table: $\text{DT}(x, y, s_1, ..., s_k) = 1$ if and only if $\text{DT}(x-1, y, s_1, ..., s_k) = 1$ or $\text{DT}(x-1, y-1, s'_1, ..., s'_k) = 1$ where $s'_i = s_i - \overrightarrow{\succ_{v(x)}}[i]$ for each $1 \le i \le k$.

RCDV. Let $(\succ_{v(1)}, ..., \succ_{v(n)})$ be a fixed order of all votes. We maintain a table $\text{DT}(x, y, s_1, s_2, ..., s_k)$, where $1 \le x \le n$, $0 \le y \le k_{\text{DV}}$ and $|s_i| \le (m-1) \cdot n$ for every $1 \le i \le k$. We define $\text{DT}(x, y, s_1, s_2, ..., s_k) = 1$ if and only if there exists $\Pi_V \subseteq \{\succ_{v(1)}, ..., \succ_{v(x)}\}$ such that $|\Pi_V| = y$ and for every $c_i \in J$ it holds that $\text{SC}(c_i) - scg(c_i, \Pi_V) = s_i$. Initially, we set $\text{DT}(1, 0, s_1, ..., s_k) = 1$ if and only if $\text{SC}(c_i) = s_i$ for every $1 \le i \le k$. In addition, $\text{DT}(1, 1, s_1, ..., s_k) = 1$ if and only if $\text{SC}(c_i) - \overrightarrow{\succ_{v(1)}}[i] = s_i$ for every $1 \le i \le k$. Moreover, every $\text{DT}(x, y, s_1, s_2, ..., s_k) = 1$ if $y > x$. We use the following relation to update the table: $\text{DT}(x, y, s_1, ..., s_k) = 1$ if and only if $\text{DT}(x-1, y, s_1, ..., s_k) = 1$ or $\text{DT}(x-1, y-1, s'_1, ..., s'_k) = 1$ for $s'_i = s_i - \overrightarrow{\succ_{v(x)}}[i]$ for each $1 \le i \le k$.

Now we describe the conditions to determine whether I' is a Yes-instance. For RCX where $X \in \{\text{AV, DV}\}$, if $Y \in \{\text{RC, RV}\}$, then I' is a Yes-instance if and only if there is an entry $\text{DT}(x, y, s_1, ..., s_k) = 1$ such that $s_i < 0$ for every $1 \le i \le k$. If $Y = \text{FO}$, let $\rhd$ be the predefined order to break the tie. Then, I' is a Yes-instance if and only if there is an entry $\text{DT}(x, y, s_1, ..., s_k) = 1$ such that (1) for every $c_i \rhd q$, $s_i < 0$; and (2) for every $q \rhd c_i$, $s_i \le 0$.

It is fairy easy to verify that the tables in the above algorithms are bounded by $O^*((mn)^k)$. Hence, the algorithms terminate in $O^*((mn)^k)$ time. □

Now we consider RCAC and RCDC. We first prove that RCAC and RCDC for Borda are in XP with respect to k. Recently, DCAC and DCDC for Borda were shown to be polynomial-time solvable [18]. Our results generalize these results.

THEOREM 8. *RCAC and RCDC for $\mu^Y \circ Borda$ are in XP with respect to k, for every $Y \in \{FO, RC, RV\}$.*

PROOF. Due to space limitation, we give only the proof for RCDC.

Let $(a_1, ..., a_k)$ be a fixed order of J. Obviously, if a candidate c is deleted, the score of every $c' \in \mathcal{C} \setminus \{c\}$ decreases by

$$|\{\succ \in \Pi_{\mathcal{V}} \mid c' \succ c\}|.$$

For each $c \in \mathcal{C} \setminus (J \cup \{q\})$ and $c' \in J$, let

$$scgap(c', c) = |\{\succ \in \Pi_{\mathcal{V}} \mid c' \succ c\}| - |\{\succ \in \Pi_{\mathcal{V}} \mid q \succ c\}|.$$

Moreover, for each candidate $c \in \mathcal{C} \setminus (J \cup \{q\})$, let $\overrightarrow{c}$ be the vector $\langle scgap(a_1, c), scgap(a_2, c), ..., scgap(a_k, c) \rangle$. Clearly, it holds that $|scgap(a_i, c)| \le n$ for every $1 \le i \le k$. We solve I' with a dynamic programming algorithm. In particular, we maintain a table $\text{DT}(x, r, \overrightarrow{s})$, where $1 \le x \le m - k - 1$, $r \le k_{\text{DC}}$, and

$\overrightarrow{s}$ is a k-dimensional integer vector with each component ranging from $-n \cdot k_{\text{DC}}$ to $n \cdot k_{\text{DC}}$. Let $(c_1, c_2, ..., c_{m-k-1})$ be any arbitrary but fixed order of the candidates in $\mathcal{C} \setminus (J \cup \{q\})$. The value of the $\text{DT}(x, r, \overrightarrow{s})$ indicates whether there exists a $B \subseteq \{c_1, c_2, ..., c_x\}$ such that $|B| = r$ and $\sum_{c \in B} scgap(a_i, c) = \overrightarrow{s}[i]$ for every $1 \leq i \leq k$, where $\overrightarrow{s}[i]$ is the ith component of $\overrightarrow{s}$. To calculate the values of the entries, we start with the initiation as follows:

1. $\text{DT}(1, 1, \overrightarrow{s}) = 1$ if and only if $\overrightarrow{s} = \overrightarrow{c_1}$;
2. $\text{DT}(1, 0, \overrightarrow{s}) = 1$ if and only if $\overrightarrow{s} = \langle 0, 0, ..., 0 \rangle$; and
3. every $\text{DT}(x, r, \overrightarrow{s}) = 1$ if $r > x$.

We use the following relation to update the table: $\text{DT}(x, r, \overrightarrow{s}) = 1$ if and only if $\text{DT}(x-1, r, \overrightarrow{s}) = 1$ or $\text{DT}(x-1, r-1, \overrightarrow{s} - \overrightarrow{c_x}) = 1$.

For each $c \in \mathcal{C}$, let $score(c, (\mathcal{C}, \Pi_{\mathcal{V}}))$ be the Borda score of c in the election $(\mathcal{C}, \Pi_{\mathcal{V}})$. Due to the definition of the table, for $Y \in \{\text{RC}, \text{RV}\}$, I' is a Yes-instance if and only if there exists $\text{DT}(x, r, \overrightarrow{s}) = 1$ such that for every $1 \leq i \leq k$,

$$score(a_i, (\mathcal{C}, \Pi_{\mathcal{V}})) - score(q, (\mathcal{C}, \Pi_{\mathcal{V}})) < \overrightarrow{s}[i].$$

For $Y = \text{FO}$, I' is a Yes-instance if and only if there is an entry $\text{DT}(x, r, \overrightarrow{s}) = 1$ such that

1. for every $a_i \in J$ with $a_i \rhd q$,
   $score(a_i, (\mathcal{C}, \Pi_{\mathcal{V}})) - score(q, (\mathcal{C}, \Pi_{\mathcal{V}})) < \overrightarrow{s}[i]$; and
2. for every $a_i \in J$ with $q \rhd a_i$,
   $score(a_i, (\mathcal{C}, \Pi_{\mathcal{V}})) - score(q, (\mathcal{C}, \Pi_{\mathcal{V}})) \leq \overrightarrow{s}[i]$;

where $\rhd$ is the pre-defined tie-breaking order.

It is fairly easy to verify that the size of the table in the above algorithm is bounded by $O^*((nm)^k)$. Therefore, the running time of the algorithm is $O^*((nm)^k)$. □

4. TIE-BREAKING SCHEMES

Our complexity results of resolute control problems do not distinguish between the tie-breaking schemes μ^{FO}, μ^{RC} and μ^{RV}. A natural question is whether the complexity is in fact independent of the tie-breaking schemes. Before we answer this question, let's recall some related work.

The complexity of various voting problems with respect to different tie-breaking schemes has been studied in the literature recently. It turned out that tie-breaking schemes may have fundamental impact on the complexity results [19, 24]. For instance, Obraztsova, Elkind and Hazon [24] revisited the polynomial-time algorithm of a manipulation problem proposed by Bartholdi, Tovey and Trick [2]. The algorithm studied in [2] takes the assumption that ties are broken in favor of the manipulators' preferred candidate. Obraztsova, Elkind and Hazon showed that the polynomial-time algorithm extends to many voting rules where ties are broken with μ^{FO} and μ^{RC}. On the other hand, they developed a polynomial-time tie-breaking scheme and showed that the manipulation problem becomes NP-hard if ties are broken with this tie-breaking scheme. Aziz et al. [1] studied a manipulation problem with respect to the tie-breaking schemes μ^{RC} and μ^{RV}, and showed that there is no direct connection between the complexity of the problems with respect to μ^{RC} and μ^{RV}. In this section, we show that analogous phenomena occurs as well in resolute control problems. For this purpose, we first study a tie-breaking scheme μ^{X3C} for which RCDC for Maximin is NP-hard.

The tie-breaking scheme μ^{X3C} resorts to the X3C problem defined as follows.

Exact 3-Set Cover (X3C)
Input: An ordered set $X = \{x_1, x_2, ..., x_{3\kappa}\}$ and a collection $S = \{s_1, s_2, ..., s_t\}$ of 3-subsets of X.
Question: Is there an $S' \subseteq S$ such that $|S'| = \kappa$ and each $x_i \in X$ appears in exactly one set of S'?

We assume that each $x_i \in X$ occurs in exactly 3 subsets of S. This does not change the complexity of the X3C problem [14]. Under this assumption, $t = 3\kappa$.

Given an ordered set $X = \{x_1, x_2, ..., x_{3\kappa}\}$, there are in total $\binom{3\kappa}{3}$ different 3-subsets of X. Moreover, the order $(x_1, ..., x_{3\kappa})$ implies a lexicographic order over all 3-subsets of X. Let $f_\kappa(i)$ be the i-th 3-subset in this order.

Now we describe the tie-breaking scheme μ^{X3C}. For each $1 \leq i \leq m$, let c_i be the i-th candidate in $\mathcal{C}$. Let $C \subseteq \mathcal{C}$ be the set of tied candidates. The tie-breaking scheme μ^{X3C} uses μ^{RC} to break the tie if one of the following cases occurs:

(1) there are no integers $\kappa > 0$ and $\ell = \binom{3\kappa}{3}$ such that $m = \ell + 2\kappa + 2$;

(2) $|C' \cap C| \neq \kappa$, where C' is the set of the first ℓ candidates of $\mathcal{C}$; or

(3)$\{c_{\ell+1}, c_{\ell+2}\} \setminus C \neq \emptyset$, i.e., both the $(\ell+1)$-th and the $(\ell+2)$-th candidate must be the tied candidates.

If none of the above cases occurs, it breaks the ties as follows: If $\{f_\kappa(i) \mid 1 \leq i \leq \ell, c_i \in C\}$ is an exact 3-set cover of $\{x_1, ..., x_{3\kappa}\}$, $c_{\ell+1}$ wins with probability 1; otherwise, $c_{\ell+2}$ wins with probability 1.

We have shown that RCDC for $\mu^Y \circ$Maximin, $Y \in \{\text{FO, RC, RV}\}$ is polynomial-time solvable (Theorem 3). Now we show that RCDC for $\mu^{\text{X3C}} \circ$ Maximin is NP-hard. McGarvey [20] proved the following lemma.

LEMMA 2. *Let $\mathcal{C}$ be a set of m candidates and ψ any function mapping from $\mathcal{C} \times \mathcal{C}$ to $\mathbb{Z}$ such that for every two distinct candidates $c, c' \in \mathcal{C}$ it holds that $\psi(c, c') = -\psi(c', c)$. Then, there is a polynomial-time algorithm to create a multiset of n votes over $\mathcal{C}$ such that n is even and for every two distinct candidates $c, c' \in \mathcal{C}$, $D(c, c') = \psi(c, c')$, where $D(c, c')$ is the number of votes ranking c above c' minus the number of votes ranking c' above c.*

THEOREM 9. *RCDC for $\mu^{X3C} \circ$ Maximin is NP-hard.*

PROOF. We prove the theorem by a reduction from the X3C problem. Let $I = (X = \{x_1, ..., x_{3\kappa}\}, S = \{s_1, ..., s_{3\kappa}\})$ be an X3C instance (assume that each x_i occurs in exactly 3 subsets of S). Let $\ell = \binom{3\kappa}{3}$. We construct an instance $(\mathcal{C}, \Pi_{\mathcal{V}}, J \subseteq \mathcal{C}, k_{\text{DC}})$ for RCDC as follows. We create in total $\ell+2+3\kappa$ candidates. First, we create an ordered set of ℓ candidates $A_{collection} = \{c_1, ..., c_\ell\}$. Then, we create two candidates $c_{\ell+1}$ and $c_{\ell+2}$. Finally, we create an ordered set of 3κ candidates $A_{selector} = \{c_{\ell+2+1}, ..., d_{\ell+2+3\kappa}\}$. Hence, we have that $\mathcal{C} = \{c_1, ..., c_{\ell+2+3\kappa}\}$, with the candidates ordered according to the indices. Let $A \subset A_{collection}$ be the set of candidates corresponding to 3-subsets in S with respect to f_κ, i.e., $A = \{c_i \in A_{collection} \mid 1 \leq i \leq \ell, f_\kappa(i) \in S\}$. For two candidates c, c', let $D(c, c') = N(c, c') - N(c', c)$, the number of votes ranking c above c' minus the number of votes ranking c' above c. We create an even number of n votes such that

- for every $c \in A_{selector}$, there are exactly $\kappa + 1$ candidates $c' \in A_{selector}$ such that $D(c, c') = -2$;
- for every $c \in A_{collection} \setminus A$, there are exactly $\kappa + 1$ candidates $c' \in A_{collection} \setminus A$ such that $D(c, c') = -2$;
- for every $c_i \in A$, $D(c_i, c_{\ell+2+i}) = -2$ and $D(c_i, c) = 0$ for every other candidate $c \in \mathcal{C} \setminus \{c_i, c_{\ell+2+i}\}$;

- $D(c_{\ell+1}, c) = 0$ for every candidate c other than $c_{\ell+1}$; and
- $D(c_{\ell+2}, c) = 0$ for every candidate c other than $c_{\ell+2}$.

Due to Lemma 2, these votes can be constructed in polynomial time. Finally, we set $k_{\text{DC}} = \kappa$ and $J = \{c_{\ell+2}\}$. Before proving the correctness of the reduction, let's summarize some useful information. First, $c_{\ell+1}$ and $c_{\ell+2}$ have Maximin score $n/2$ and every other candidate has Maximin score at most $n/2-1$. As $m \neq \ell+2\kappa+2$, due to the tie-breaking scheme, $c_{\ell+2}$ has a positive probability to win. Moreover, as for every $c \in A_{selector} \cup A_{collection} \setminus A$ there are $\kappa+1$ candidates c' with $D(c, c') = -2$, every candidate in $A_{selector} \cup A_{collection} \setminus A$ still has Maximin score at most $n/2-1$ after deleting any κ candidates. Furthermore, a candidate $c_i \in A$ has Maximin score $n/2$ in the final election if and only if the candidate $c_{\ell+2+i}$ is deleted.

Now we prove the correctness. Assume that S' is an exact 3-set cover. Consider deleting all candidates in $\{c_{\ell+2+i} \mid f_\kappa(i) \in S'\}$. Then, according to the above discussion, all candidates $c_i \in A$ such that $f_\kappa(i) \in S'$ has Maximin score $n/2$, and hence, tied with $c_{\ell+1}$ and $c_{\ell+2}$. Then, due to the definition of μ^{X3C}, $c_{\ell+1}$ wins with probability 1. Thus, the instance of RCDC is a Yes-instance.

It remains to prove opposite direction. As $c_{\ell+2}$ has Maximin score $n/2$ no matter which κ candidates are deleted, due to μ^{X3C}, to prevent $c_{\ell+2}$ from having a positive probability to win, we have to delete exactly κ candidates, so that in the final election there are exactly $\ell+2+2\kappa$ candidates (Condition (1)). Moreover, these deleted candidates must be all from the last 3κ candidates in $\mathcal{C}$, since otherwise, the $(\ell+2)$-th candidate in the final election would be someone in the last 3κ candidates in the original election which is not a winner as discussed above, and hence, due to μ^{X3C}, all tied candidates have positive probability to win including $c_{\ell+2}$ (Condition (3)). Furthermore, after deleting these candidates, there must be a subset $A' \subset A_{collection}$ such that $|A'| = 3\kappa$ (Condition (2)), each candidate in A' ties with $c_{\ell+1}$ and $c_{\ell+2}$, and $\{f_\kappa(i) \mid c_i \in A', 1 \leq i \leq \ell\}$ is an exact 3-set cover. As discussed above, every candidate in $A_{collection} \setminus A$ has Maximin score $n/2-1$ no matter which κ candidates are deleted. Hence, none of the candidates in $A_{collection} \setminus A$ ties with $c_{\ell+1}$ and $c_{\ell+2}$ in the final election. This implies that A' is a subset of A. As a result, for every $c_i \in A'$, $f_\kappa(i) \in S$, implying that I is a Yes-instance. □

One may still wonder whether the three natural tie-breaking schemes μ^{FO}, μ^{RC} and μ^{RV} also have impact on the complexity results. We clear up the confusion by the following theorem.

THEOREM 10. *There exists a non-Pareto optimal voting correspondence φ such that RCAV for voting rules $\mu^Y \circ \varphi$ for $Y \in \{FO, RC\}$ is NP-hard but for $Y = RV$ is polynomial-time solvable.*

PROOF. Consider the following voting correspondence φ. Let a and b be the two lexicographically smallest candidates. If a is ranked in the first place in all votes and b is the unique Borda winner among all the other remaining candidates, then a wins as the unique winner; otherwise, a and b win as co-winners. Then, CCAV for φ is NP-hard (with the distinguished candidate to be a) since it is equivalent to determining whether b can be made the unique Borda winner in the election without a by adding k_{AV} unregistered votes. Due to Lemma 1, RCAV for $\mu^{\text{FO}} \circ \varphi$ and $\mu^{\text{RC}} \circ \varphi$ are NP-hard. However, RCAV for $\mu^{\text{RV}} \circ \varphi$ is polynomial-time solvable. Assume that $J \neq \mathcal{C}$ and $|\{a, b\} \cap J| \leq 1$ (otherwise, the given instance is a No-instance). We distinguish between three cases. Case 1: $a \in J$. In this case, if there is a registered vote that ranks a above b, return "No"; otherwise, return "Yes" since $\emptyset$ is a solution. Case 2: $b \in J$. In this case, if there is a registered vote which ranks b above a, then return "No"; otherwise, return "Yes", since $\emptyset$ is a solution. Case 3: $\{a, b\} \cap J = \emptyset$. In this case, we directly return "Yes", since no candidate other than a and b can win the election. □

Recall that the statement for μ^{RV} in Lemma 1 requires the voting correspondence to be Pareto optimal. Theorem 10 implies that this requirement is essential in Lemma 1.

5. CONCLUSION

We studied the complexity of the resolute control problems for Borda, Maximin and Copeland$^\alpha$, where $0 \leq \alpha \leq 1$, with ties being broken with μ^{FO}, μ^{RC} and μ^{RV}. In these problems, we are given an election with a set J of distinguished candidates, and an external agent wants to prevent all distinguished candidates from having a positive probability to win. Resolute control problems are natural generalizations of destructive control problems that have been extensively studied in the literature. We first identified polynomial-time solvable resolute control problems and NP-hard resolute control problems. Then, we further investigated the parameterized complexity of the NP-hard resolute control problems with respect to the number of distinguished candidates. In particular, for many of them we either developed FPT-algorithms or XP-algorithms. For instance, for RCDC for Maximin, we developed a single exponential time algorithm with running time $O^*(2^k)$. See Table 1 for a summary of our complexity results.

In addition, we studied the impact of tie-breaking schemes on the complexity of resolute control problems. We showed that there are tie-breaking schemes with respect to which the complexity of the resolute control problems differs. Nevertheless, it is important to point out that either the tie-breaking schemes or the voting correspondences involved in our results are made artificially (Theorems 9 and 10). It remains as an intriguing open question whether there are commonly used voting correspondences and natural tie-breaking schemes for which the complexity of resolute control problems differs. It is also important to point out that even though the three natural tie-breaking schemes μ^{FO}, μ^{RC} and μ^{RV} do not have impact on the complexity of resolute control problems for Borda, Maximin and Copland$^\alpha$ as shown in this paper, we have to take them into account when developing FPT- or XP-algorithms for the resolute control problems.

There remain several open questions. For instance, we do not know whether the resolute control problems for Borda are FPT. For FPT problems studied in this paper, a challenging task would be to develop polynomial kernels, or prove that they do not admit a polynomial kernel. Recall that a *kernelization* of a parameterized problem Q is a polynomial-time algorithm that takes as input an instance of Q and outputs an equivalent new instance of Q whose size is bounded by a function of the parameter. Kernelization is one of the most significant approaches to deal with FPT problems. We refer to [16] for a comprehensive survey on kernelization.

Another direction for future research would be to study the resolute control problems in restricted elections. In particular, for pa-NP-hard problems studied in this paper, it is intriguing to know if they become FPT if the voters' preferences are subject to some combinatorial properties. Recently, complexity of voting problems in elections with restricted preferences has received a considerable amount of attention, see, e.g., [8, 29, 30, 32, 33].

Acknowledgments

The authors would like to thank the anonymous referees of AAMAS 2017 for their helpful comments. This work is supported in part by the National Natural Science Foundation of China (NSFC) under grants 61420106009 and 61232001.

REFERENCES

[1] H. Aziz, S. Gaspers, N. Mattei, N. Narodytska, and T. Walsh. Ties matter: Complexity of manipulation when tie-breaking with a random vote. In *AAAI*, pages 74–80, 2013.

[2] J. J. Bartholdi III, C. A. Tovey, and M. A. Trick. The computational difficulty of manipulating an election. *Soc. Choice. Welfare.*, 6(3):227–241, 1989.

[3] J. J. Bartholdi III, C. A. Tovey, and M. A. Trick. How hard is it to control an election? *Math. Comput. Model.*, 16(8-9):27–40, 1992.

[4] D. Baumeister, G. Erdélyi, E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. *Computational Aspects of Approval Voting*, chapter 10, pages 199–251. Handbook on Approval Voting. Springer Berlin Heidelberg, 2010.

[5] R. Bredereck, P. Faliszewski, R. Niedermeier, and N. Talmon. Large-scale election campaigns: Combinatorial shift bribery. *J. Artif. Intell. Res.*, 55:603–652, 2016.

[6] V. Conitzer, T. Sandholm, and J. Lang. When are elections with few candidates hard to manipulate? *J. ACM.*, 54(3), 2007. Article 14.

[7] P. Dey, N. Misra, and Y. Narahari. Frugal bribery in voting. In *AAAI*, pages 2466–2472, 2016.

[8] G. Erdélyi, M. Lackner, and A. Pfandler. Manipulation of k-approval in nearly single-peaked electorates. In *ADT*, pages 71–85, 2015.

[9] G. Erdélyi, M. Nowak, and J. Rothe. Sincere-strategy preference-based approval voting fully resists constructive control and broadly resists destructive control. *Math. Log. Q.*, 55(4):425–443, 2009.

[10] G. Erdélyi, C. Reger, and Y. Yang. The Complexity of Bribery and Control in Group Identification. In *AAMAS*, 2017. to appear.

[11] P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra. How hard is bribery in elections? *J. Artif. Intell. Res.*, 35:485–532, 2009.

[12] P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra. Multimode control attacks on elections. *J. Artif. Intell. Res.*, 40:305–351, 2011.

[13] P. Faliszewski, E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. Llull and Copeland voting computationally resist bribery and constructive control. *J. Artif. Intell. Res.*, 35:275–341, 2009.

[14] T. F. Gonzalez. Clustering to minimize the maximum intercluster distance. *Theoret. Comput. Sci.*, 38:293–306, 1985.

[15] E. Hemaspaandra, L. A. Hemaspaandra, and J. Rothe. Anyone but him: The complexity of precluding an alternative. *Artif. Intell.*, 171(5-6):255–285, 2007.

[16] S. Kratsch. Recent developments in kernelization: A survey. *Bulletin of the EATCS*, 113:58–97, 2014.

[17] H. W. Lenstra. Integer programming with a fixed number of variables. *Math. Oper. Res.*, 8(4):538–548, 1983.

[18] A. Loreggia, N. Narodytska, F. Rossi, K. B. Venable, and T. Walsh. Controlling elections by replacing candidates or votes. In *AAMAS*, pages 1737–1738, 2015.

[19] N. Mattei, N. Narodytska, and T. Walsh. How hard is it to control an election by breaking ties? In *ECAI*, pages 1067–1068, 2014.

[20] D. McGarvey. A theorem on the construction of voting paradoxes. *Econometrica.*, pages 608–610, 1953.

[21] R. Meir, A. D. Procaccia, and J. S. Rosenschein. A broader picture of the complexity of strategic behavior in multi-winner elections. In *AAMAS*, pages 991–998, 2008.

[22] R. Meir, A. D. Procaccia, J. S. Rosenschein, and A. Zohar. Complexity of strategic behavior in multi-winner elections. *J. Artif. Intell. Res.*, 33:149–178, 2008.

[23] S. Obraztsova and E. Elkind. Optimal manipulation of voting rules. In *AAAI*, pages 2141–2147, 2012.

[24] S. Obraztsova, E. Elkind, and N. Hazon. Ties matter: Complexity of voting manipulation revisited. In *AAMAS*, pages 71–78, 2011.

[25] S. Obraztsova, Y. Zick, and E. Elkind. On manipulation in multi-winner elections based on scoring rules. In *AAMAS*, pages 359–366, 2013.

[26] A. D. Procaccia, J. S. Rosenschein, and A. Zohar. Multi-winner elections: Complexity of manipulation, control and winner-determination. In *IJCAI*, pages 1476–1481, 2007.

[27] N. F. Russell. Complexity of control of Borda count elections. Master's thesis, Rochester Institute of Technology, 2007.

[28] W. D. Smith. Descriptions of single-winner voting systems. http://m-schulze.9mail.de/votedesc.pdf, 2006.

[29] Y. Yang. Manipulation with bounded single-peaked width: A parameterized study. In *AAMAS*, pages 77–85, 2015.

[30] Y. Yang. On the complexity of Borda control in single-peaked elections. In *AAMAS*, 2017. To appear.

[31] Y. Yang and D. Dimitrov. How Hard Is It to Control A Group? *COMSOC*, 2016. https://arxiv.org/abs/1606.03366

[32] Y. Yang and J. Guo. The control complexity of r-Approval: from the single-peaked case to the general case. In *AAMAS*, pages 621–628, 2014.

[33] Y. Yang and J. Guo. Controlling elections with bounded single-peaked width. In *AAMAS*, pages 629–636, 2014.

[34] Y. Yang, Y. R. Shrestha, and J. Guo. How hard is bribery with distance restrictions? In *ECAI*, pages 363–371, 2016.