# On the Complexity of Destructive Bribery in Approval-Based Multi-winner Voting

Yongjie Yang
Chair of Economic Theory, Faculty of Human and Business Sciences, Saarland University
Saarbrücken, Saarland, Germany
yyongjiecs@gmail.com

## ABSTRACT

A variety of constructive manipulation, control, and bribery for approval-based multi-winner voting have been extensively studied very recently. However, their destructive counterparts seem to be less studied in the literature so far. This paper aims to fill this gap by exploring the complexity of several destructive bribery problems under five prestigious approval-based multi-winner voting rules. Generally speaking, these problems are to determine if a number of given candidates can be excluded from any winning committees by performing a series of modification operations yet without exceeding a given budget. We consider five operations. We offer a complete landscape of the complexity of the problems studied in this paper, and for NP-hard problems we study their parameterized complexity with respect to meaningful parameters.

## KEYWORDS

destructive bribery, satisfaction approval voting, Chamberlin-Courant approval voting, proportional approval voting, multi-winner voting, parameterized complexity, FPT, W[1]-hard, NP-hard

**ACM Reference Format:**

Yongjie Yang. 2020. On the Complexity of Destructive Bribery in Approval-Based Multi-winner Voting. In *Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), Auckland, New Zealand, May 9–13, 2020,* IFAAMAS, 9 pages.

## 1 INTRODUCTION

After more than two decades of extensive study on the complexity of single-winner voting problems, the computational social choice community shifted their main focus to multi-winner voting very recently. Many variants of manipulation, control, and bribery problems for approval-based multi-winner voting rules (ABM rules for short) have been studied from the complexity point of view (see e.g., [2, 23, 41, 47]). However, these works are mainly concerned with the constructive model of these problems where, in general, one is interested in making a single distinguished candidate a winner, or making a committee a winning committee. The destructive counterparts of these problems seem not to have been widely studied in the literature so far. Aiming at filling this gap, we propose several destructive bribery problems for ABM rules and study their complexity and parameterized complexity. Our problems are defined to model the applications where an election attacker (or briber) wants to preclude multiple distinguished candidates from winning by making some changes of the votes (or bribing some voters so that they change their votes in a certain way) while not exceeding his/her budget. The behavior of the attacker may be motivated by, for example, that these distinguished candidates are his/her rivals (e.g., these candidates have completely different political views from the attacker), or the attacker wants to make them lose in order to increase the chance of making his/her liked candidates win. We consider five particular bribery operations classified into two classes: *atomic operations* and *vote-level operations*. Approval addition (AppAdd) and approval deletion (AppDel) are atomic operations where each single AppAdd/AppDel means to add/delete one candidate into/from the set of approved candidates of some vote. Vote-level change (VC), vote-level addition change (VAC), and vote-level deletion change (VDC) are vote-level operations where each single operation respectively means to change a vote in any possible way, change a vote by adding some candidates into the set of the approved candidates, and change a vote by deleting some candidates from the set of approved candidates. Each bribery problem is associated with an operation type and the attacker can perform at most a given number of single operations of the same type. For vote-level operation problems, we also introduce an integer distance bound $r$ and assume that each vote can be only changed into another one which has Hamming-distance at most $r$ from the vote. This parameter models the scenarios where voters do not want to deviate too much from their true preferences. We point out that bribery problems with distance restrictions have been studied in the setting of single-winner voting recently [7, 16, 44].

We study these problems under five widely-studied ABM rules, namely, approval voting (AV), satisfaction approval voting (SAV), net-satisfaction approval voting (NSAV), Chamberlin-Courant approval voting (CCAV), and proportional approval voting (PAV). We obtain the complexity of all problems considered in the paper. Many of our NP-hardness results hold even in very special cases. For NP-hardness results, we also explore how numerous meaningful parameters shape the parameterized complexity of these problems. We obtain both fixed-parameter tractability (FPT) results and W[1]-hardness results.

**Related Work.** Our work is clearly related to the pioneering works of Bartholdi et al. [4–6] where numerous strategic single-winner voting problems have been studied from the complexity point of view, motivated by that complexity can be regarded as a barrier against strategic behavior.[1] Since their seminal work, investigating the complexity of many single-winner voting problems, particularly of strategic problems in both constructive model and destructive model, has been dominating the advance of computational social

*Proc. of the 19th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2020), B. An, N. Yorke-Smith, A. El Fallah Seghrouchni, G. Sukthankar (eds.), May 9–13, 2020, Auckland, New Zealand.* 

[1] It should be pointed out that several recent studies have shown that many computationally hard voting problems may be solved efficiently for practical elections.

choice. However, the research on the complexity of multi-winner voting problems had lagged behind with only a few related papers being published [9, 29, 31] until the work of Aziz et al. [2]. In particular, Aziz et al. [2] studied the complexity of the winners determination and several constructive manipulation problems for ABM rules. Their work largely sparked the extensive and intensive study of the complexity of voting problems for ABM rules. Among all these studies, the following works are most related to ours. Meir et al. [31] studied both constructive and destructive manipulation and control but mainly for ranking-based multi-winner voting rules. Faliszewski et al. [23] studied various constructive bribery problems for ABM rules. Particularly, they also studied the operations AppAdd and AppDel. Bredereck et al. [12] studied constructive shift bribery for both ranking-based multiwinner voting rules and ABM rules. Very recently, Yang [41] investigated the complexity of constructive manipulation and control problems for ABM rules. However, in these problems there is only one distinguished candidate which is wanted to be included in the winning committee by the election attackers, but we consider multiple distinguished candidates who are desired by the attackers not to be in any winning committees. It should be also pointed out that destructive strategic voting problems with multiple distinguished candidates have been explored in single-winner voting problems [45] and in the setting of group identification [22, 43]. Aziz et al. [2] also studied a constructive manipulation problem where there are exactly $k$ distinguished candidates, and the goal is to make them exactly the winners. In contrast, we study destructive bribery problems and the number of distinguished candidates is not necessarily to be the number of winners.

The destructive bribery problems studied in this paper are more or less related to the concept of robustness of multi-winner voting rules which is concerned with the minimum amount of changes to alter the winning committees. In particular, Gawron and Faliszewski [26] recently studied the complexity of determining how many operations are needed to change the set of winning committees in the setting of approval-based voting. They considered the adding, deleting, and replacing operations.

## 2 PRELIMINARIES

In approval-based voting, each voter is asked to report a subset of candidates who s/he approves. Formally, an approval-based election is a tuple $(C, V)$ where $C$ is a set of candidates and $V$ is a multiset of votes, each vote is cast by a voter and is defined as a subset of candidates consisting of all candidates approved by the voter. In this paper, a subset of candidates is called a committee, and a subset of exactly $k$ candidates is called a $k$-committee. A multi-winner voting rule $f$ assigns to each election $(C, V)$ and an integer $k \leq |C|$ a collection of $k$-committees, called the winning $k$-committees under $f$. For a candidate $c \in C$, let $V(c) = \{v \in V : c \in v\}$ be the multiset of votes approving $c$.

In this paper, we study the rules AV, SAV, NSAV, CCAV, and PAV. In these rules, each vote offers a certain score to each committee, and winning $k$-committees are those having the maximum total scores received from all votes. These rules differ only at how the scores are defined. The scores of a committee $w \subseteq C$ with respect to these rules are summarized in Table 1.

**Table 1: Scores of five multi-winner voting rules.**

| rules | total scores of $w \subseteq C$ in an election $(C, V)$ |
|---|---|
| AV | $\sum_{v \in V} \|v \cap w\|$ |
| SAV | $\sum_{v \in V, v \neq \emptyset} \frac{\|v \cap w\|}{\|v\|}$ |
| NSAV | $\sum_{v \in V, v \neq \emptyset} \frac{\|v \cap w\|}{\|v\|} - \sum_{v \in V, v \neq C} \frac{\|w \setminus v\|}{\|C\|-\|v\|}$ |
| CCAV | $\|\{v \in V : v \cap w \neq \emptyset\}\|$ |
| PAV | $\sum_{v \in V, v \cap w \neq \emptyset} \sum_{i=1}^{\|v \cap w\|} \frac{1}{i}$ |

In AV, each voter gives 1 point to every candidate s/he approves. In SAV, each voter has a fixed 1 point which is equally distributed among her/his approved candidates. NSAV takes a step further by allowing voters to express their dissatisfaction with their disapproved candidates. Particularly, in addition to the fixed 1 point like SAV, each voter also equally distributes $-1$ point among all her/his disapproved candidates. The AV/SAV/NSAV score of a committee is the sum of the total scores of its members. SAV and NSAV were respectively proposed by Brams and Kilgour [10] and Kilgour and Marshall [28]. In CCAV voting, a voter is satisfied by a committee if at least one of her/his approved candidates is included in the committee. This rule selects $k$-committees satisfying the maximum number of voters. CCAV is a special case of a class of rules studied in [14], and was suggested by Thiele [37]. In PAV, each committee $w$ receives $1 + \frac{1}{2} + \cdots + \frac{1}{|v \cap w|}$ points from each vote $v$ such that $v \cap w \neq \emptyset$. PAV was first mentioned in the work of Thiele [37]. A significant difference among these rules is that calculating a winning $k$-committee is NP-hard for CCAV and PAV but polynomial-time solvable for AV, SAV, and NSAV [2].

For each $f \in \{\text{AV}, \text{SAV}, \text{NSAV}, \text{CCAV}, \text{PAV}\}$ and a committee $w \subseteq C$ in an election $(C, V)$, let $f_{(C,V)}(w)$ denote the $f$ score of $w$ received from all votes in $V$. For a singleton committee $\{c\}$ where $c \in C$, we write $f_{(C,V)}(c)$ for $f_{(C,V)}(\{c\})$ for notion simplicity. We omit the subindex from the notion if it is clear from the context which election is considered.

We study five destructive bribery problems characterized by five modification operations, including two atomic operations and three vote-level change operations. The two atomic operations are defined as follows.

**Approval addition (AppAdd)** A single AppAdd operation on some vote $v \in V$ such that $v \neq C$ means that we extend $v$ by adding exactly one candidate in $C \setminus v$ into $v$.

**Approval deletion (AppDel)** A single AppDel operation on some vote $v \in V$ such that $v \neq \emptyset$ means that we remove one candidate from $v$.

Let $f$ be an ABM rule. Let $X$ be an atomic operation.

| Destructive $X$ Bribery for $f$ (DXB-$f$) | |
|---|---|
| **Input:** | An election $(C, V)$, a nonempty subset $J \subseteq C$ of distinguished candidates, and two positive integers $k \leq \|C\|$ and $\ell$. |
| **Question:** | Can we perform at most $\ell$ many $X$ operations in $(C, V)$ so that none of $J$ is in any winning $k$-committees under $f$? |

Different from atomic operations, each single vote-level operation changes one vote in some specific way.

**Vote change (VC)** A single VC operation on some vote $v$ means to change $v$ into another vote which can be any subset of candidates.

**Vote addition change (VAC)** A single VAC operation on some vote $v$ such that $v \neq C$ means that we add some (one or more) candidates from $C \setminus v$ into $v$.

**Vote deletion change (VDC)** A single VDC operation on some vote $v$ such that $v \neq \emptyset$ means that we remove some (one or more) candidates from $v$.

To generalize our study as much as possible, we consider the distance-bounded bribery model where we are given an additional nonnegative integer $r$, and request that the Hamming distance between a bribed vote $v$ and the new vote after a single vote-level operation on $v$ is at most $r$. Recall that the Hamming distance between two votes $v \subseteq C$ and $v' \subseteq C$ is $|v \setminus v'| + |v' \setminus v|$. When $r$ is the number of candidates, this restriction completely fades out. Many of our NP-hardness results hold even when $r$ is a small constant.

For a vote-level operation $Y$ defined above and a nonnegative integer distance bound $r$, we study the following problem.

| $r$-Bounded Destructive $Y$ Bribery for $f$ ($r$-DYB-$f$) | |
|---|---|
| **Input:** | An election $(C, V)$, a nonempty subset $J \subseteq C$ of distinguished candidates, and two positive integers $k \leq |C|$ and $\ell \leq |V|$. |
| **Question:** | Is there a subset $V' \subseteq V$ of at most $\ell$ votes such that we can perform a single $Y$ operation on every vote in $V'$ such that the Hamming distance between the vote after the operation and the original vote is at most $r$ and, moreover, after all these $|V'|$ operations none of $J$ is in any winning $k$-committees under $f$? |

We assume the reader is familiar with the basics in computational complexity, parameterized complexity, and graph theory, and we refer to [17, 38, 39] for consultation. Our hardness results in the paper are based on reductions from the following problems.

| Restricted Exact Cover by Three Sets (RX3C) | |
|---|---|
| **Input:** | A universe $A$ of cardinality $3\kappa$ for some positive integer $\kappa$, and a collection $\mathcal{H}$ of subsets of $A$ such that each subset in $\mathcal{H}$ is of cardinality 3, and each element in $A$ appears in exactly three elements of $\mathcal{H}$. |
| **Question:** | Is there an exact set cover of $A$, i.e., a subcollection $\mathcal{H}' \subseteq \mathcal{H}$ such that every element in $A$ appears in exactly one element of $\mathcal{H}'$? |

It is known that the RX3C problem is NP-hard [27]. Note that for every RX3C instance $(A, \mathcal{H})$, it holds that $|\mathcal{H}| = |A| = 3\kappa$, and each solution $\mathcal{H}'$ is of cardinality $\kappa$.

An independent set of a graph is a subset of vertices whose induced subgraph contains no edges.

| $\kappa$-Independent Set | |
|---|---|
| **Input:** | A graph $G$ and a positive integer $\kappa$. |
| **Question:** | Does $G$ have an independent set of size $\kappa$? |

A clique in a graph is a subset of vertices whose induced subgraph is complete (i.e., there is an edge between every two vertices).

| $\kappa$-Clique | |
|---|---|
| **Input:** | A graph $G$ and an integer $\kappa$. |
| **Question:** | Does $G$ have a clique of size $\kappa$? |

It is well-known that both the $\kappa$-Independent Set and the $\kappa$-Clique problems are W[1]-hard with respect to the parameter $\kappa$ [18]. Moreover, both problems remain W[1]-hard even when restricted to regular graphs [13].

For a graph $G = (U, A)$ and a subset $U' \subseteq U$, $G[U']$ denotes the subgraph of $G$ induced by $U'$. For a positive integer $i$, let $[i] = \{j \in \mathbb{N} : 1 \leq j \leq i\}$ be the set of all positive integers no greater than $i$.

Due to space limitations, we omit several proofs, and we mark the theorems whose proofs are omitted by $\star$. We also omit the analysis of the exact running times of polynomial-time algorithms obtained in this paper. A full version of this paper is available at https://arxiv.org/pdf/2002.00836.pdf.

## 3 NP-HARD WINNERS DETERMINATION RULES

We start our exploration with CCAV and PAV. Unlike other rules studied in this paper, for CCAV and PAV even Winners Determination is NP-hard [2]. However, these rules are still interesting due to at least the following reasons. First, they satisfy certain proportional properties which are failed by other rules studied in the paper [1, 24, 34]. Second, many FPT-algorithms and competitive approximation algorithms have been reported for calculating winners under these rules [2, 35, 36, 46]. Polynomial-time algorithms for restricted domains have also been derived [33, 42].

We show that the bribery problems defined in this paper are W[1]-hard under CCAV and PAV with respect to the size of the winning committees even when there is only one single distinguished candidate ($|J| = 1$), the budget is 0 ($\ell = 0$), and every voter approves at most two candidates. In fact, in this special case, we have exactly the following problem: given an election, a distinguished candidate $p$, and an integer $k$, determine if $p$ is not included in any winning $k$-committees under some rule $f$. We call this special case Non-Winner Determination for $f$ (NWD-$f$). Our W[1]-hardness result is via reductions from the $\kappa$-Independent Set problem restricted to regular graphs.

Theorem 3.1. *NWD-CCAV is W[1]-hard with respect to the parameter $k$, even when every voter approves at most two candidates.*

Proof. Let $(G = (U, A), \kappa)$ be an instance of the $\kappa$-Independent Set problem where every vertex in $G$ has degree exactly $d$ for some positive integer $d$. We create an instance of NWD-CCAV, denoted by $((C, V), p \in C, k)$, as follows. For each vertex $u \in U$ in $G$, we create one candidate $c(u)$. In addition, we create a distinguished candidate $p$. So, we have $C = \{c(u) : u \in U\} \cup \{p\}$. The multiset $V$ of votes consists of the following votes. For each edge $\{u, u'\} \in A$, we create one vote $v(u, u') = \{c(u), c(u')\}$. In addition, we create $d - 1$ votes each of which approves only the distinguished candidate $p$. Finally, we set $k = \kappa$. Now we show the correctness.

($\Rightarrow$) Suppose that $G$ has an independent set of size $\kappa$. Then, every $k$-committee corresponding to an independent set of size $k$ satisfies the maximum number of $\kappa \cdot d$ votes. However, every $k$-committee containing the distinguished candidate $p$ is able to satisfy at most

**Table 2: A summary of the complexity of destructive bribery for ABM rules. Here, "P" means polynomial-time solvable. Parameterized complexity results are with respect to the parameters enclosed in the parentheses next to them. A bracket next to some parameter means that the result with respect to this parameter holds even when the conditions in the brackets are fulfilled.**

| | AppAdd | AppDel | $r$-VC | $r$-VAC | $r$-VDC |
|---|---|---|---|---|---|
| AV | P | P | NP-h ($k = 1 \wedge r \geq 4$)<br>W[1]-h ($\ell, k$) [$\lvert J\rvert = 1 \wedge r \geq 3$]<br>FPT ($m, n$ [$r = m$]) | W[1]-h ($\ell, k$) [$\lvert J\rvert = 1 \wedge r \geq 2$]<br>FPT ($m, n$ [$r = m$])<br>P ($k = 1$) | NP-h ($k = 1 \wedge r \geq 3$)<br>FPT ($\lvert J\rvert, n$)<br>P ($r$=1) |
| SAV | NP-h ($k = 1$)<br>FPT ($m$) | NP-h ($k = 1$)<br>W[1]-h ($\ell, k$) [$\lvert J\rvert = 1$]<br>FPT ($m$) | NP-h ($k = 1 \wedge r \geq 4$)<br>W[1]-h ($\ell, k$) [$\lvert J\rvert = 1 \wedge r \geq 1$]<br>FPT ($m$) | NP-h ($k = 1 \wedge r \geq 1$)<br>FPT ($m$) | NP-h ($k = 1 \wedge r \geq 3$)<br>W[1]-h ($\ell, k$) [$\lvert J\rvert = 1 \wedge r \geq 1$]<br>FPT ($m$) |
| NSAV | NP-h ($k = 1$)<br>FPT ($m$) | NP-h ($k = 1$)<br>W[1]-h ($\ell, k$) [$\lvert J\rvert = 1$]<br>FPT ($m$) | NP-h ($k = 1 \wedge r \geq 4$)<br>W[1]-h ($\ell, k$) [$\lvert J\rvert = 1 \wedge r \geq 1$]<br>FPT ($m$) | NP-h ($k = 1 \wedge r \geq 1$)<br>FPT ($m$) | NP-h ($k = 1 \wedge r \geq 3$)<br>W[1]-h ($\ell, k$) [$\lvert J\rvert = 1 \wedge r \geq 1$]<br>FPT ($m$) |
| CCAV | W[1]-h ($k$ [$\lvert J\rvert = 1 \wedge \ell = 0$])<br>FPT ($m$) | | W[1]-h ($k$ [$\lvert J\rvert = 1 \wedge \ell = 0 \wedge r \geq 0$])<br>FPT ($m$) | | |
| PAV | W[1]-h ($k$ [$\lvert J\rvert = 1 \wedge \ell = 0$])<br>FPT ($m$) | | W[1]-h ($k$ [$\lvert J\rvert = 1 \wedge \ell = 0 \wedge r \geq 0$])<br>FPT ($m$) | | |

$(\kappa - 1) \cdot d + (d - 1) = \kappa \cdot d - 1$ votes. Therefore, $p$ cannot be included in any winning $k$-committees.

($\Leftarrow$) Suppose that $G$ does not have any independent set of size $\kappa$. We claim that there exists at least one winning $k$-committee which contains the distinguished candidate $p$. If $p$ is included in all winning $k$-committees, we are done. Assume that there is a winning $k$-committee $w$ which contains only candidates corresponding to a set of $k$ vertices in $G$. As $G$ does not have an independent set of size $\kappa$, there exists at least one edge $\{u, u'\}$ in $G$ such that both $c(u)$ and $c(u')$ are in $w$. In addition, as there are exactly $d - 1$ votes who approve only the distinguished candidate $p$, it holds that the committee $w' = (w \setminus \{c(u)\}) \cup \{p\}$ satisfies at least $\mathsf{CCAV}(w) - (d - 1) + (d - 1) = \mathsf{CCAV}(w)$ votes, implying that $w'$ is also a winning $k$-committee. □

For PAV, we can obtain the same result.

Theorem 3.2 ($\star$). *NWD-PAV is W[1]-hard with respect to the parameter k, even when every voter approves at most two candidates.*

The above theorems give us the following corollary.

Corollary 3.3. *For $f \in \{CCAV, PAV\}$, the problems DAppAddB-f, DAppDelB-f, r-DVCB-f, r-DVACB-f, and r-DVDCB-f are W[1]-hard with respect to the parameter k. These hold even when $|J| = 1$, the budget of the briber is $\ell = 0$, and every voter approves at most two candidates. For r-DVCB-f, r-DVACB-f, and r-DVDCB-f, the W[1]-hardness holds for all $r \geq 0$.*

## 4 POLYNOMIAL-TIME WINNERS DETERMINATION RULES

In this section, we investigate destructive bribery for AV, SAV, and NSAV whose Winners Determination problem is polynomial-time solvable. First, we study a relation between SAV and NSAV elections which enables us to derive hardness results for NSAV from those for SAV. Assume that we have a hardness result for SAV via a reduction where an election is created. Then, to show the hardness for NSAV, we add a large set of dummy candidates who are never approved by any voter (and none of them is a distinguished candidate). The large quantity of the dummy candidates ensures that the NSAV scores of candidates are dominated by their SAV scores, in the sense that a candidate has a greater/smaller SAV score than that of another candidate if and only if the former has a greater/smaller NSAV score than that of the latter in the election after adding all dummy candidates.

Lemma 4.1. *Let $(C, V)$ be an election where $m = |C| \geq 2$ and $n = |V|$. Let $D$ be a set of at least $n \cdot m^2$ candidates disjoint from $C$. Then, for every two candidates $c$ and $c'$ in $C$, it holds that $\mathsf{SAV}_{(C,V)}(c) > \mathsf{SAV}_{(C,V)}(c')$ if and only if $\mathsf{NSAV}_{(C \cup D,V)}(c) > \mathsf{NSAV}_{(C \cup D,V)}(c')$.*

Proof. Observe that if two candidates $c, c' \in C$ have different SAV scores in $(C, V)$, then the absolute value of their score gap is at least $\frac{1}{m-1} - \frac{1}{m} = \frac{1}{m \cdot (m-1)}$. In the election $(C \cup D, V)$, candidates in $D$ are not approved by any vote in $V$. Therefore, the NSAV score of a candidate $c \in C$ in $(C \cup D, V)$ is its SAV score in $(C, V)$ minus $\sum_{v \in V, c \notin v} \frac{1}{m+|D|-|v|}$. Because $|D| \geq n \cdot m^2$ and $|v| \leq m - 1$, it holds that $\sum_{v \in V, c \notin v} \frac{1}{m+|D|-|v|} < \frac{1}{m \cdot (m-1)}$. The lemma follows. □

All hardness results for NSAV in this paper can be obtained by modifications of the reductions for the same problems under SAV by adding dummy candidates as discussed above. Lemma 4.1 ensures the correctness of the reduction for NSAV.

In the following, we divide our discussions into several subsections each of which is devoted to a concrete bribery problem.

### 4.1 Approval Addition

In this subsection, we study the atomic operation AppAdd. We show that among the five rules, AV is the only one which admits a polynomial-time algorithm.

Theorem 4.2. *DAppAddB-AV is polynomial-time solvable.*

Proof. Let $I = ((C, V), J \subseteq C, k, \ell)$ be a DAppAddB-AV instance. Let $m$ and $n$ denote the number of candidates and the number of votes, respectively. Consider first the case where there exists one candidate in $J$ which is included in all votes. In this case, we directly conclude that the given instance is a No-instance. Therefore, in the following let us assume that the above case does not occur. We derive an algorithm as follows. First, we calculate the AV scores of all candidates, and find a candidate $c^*$ in $J$ such that $\mathsf{AV}(c^*) \geq \mathsf{AV}(c)$ for all $c \in J$. Let $C^>(c^*) = \{c \in C \setminus J : \mathsf{AV}(c) > \mathsf{AV}(c^*)\}$ be the set of all nondistinguished candidates who have strictly higher AV scores than that of $c^*$. If $|C^>(c^*)| \geq k$, we conclude that $I$ is a Yes-instance. Assume that this is not the case. As adding candidates into votes never decreases AV scores of any candidates, and it is optimal to never add distinguished candidates into any vote, the question is now whether we can perform at most $\ell$ AppAdd operations so that at least $k$ candidates in $C \setminus J$ have AV scores at least $\mathsf{AV}(c^*) + 1$. For each candidate $c \in C \setminus (J \cup C^>(c^*))$, let $\mathsf{diff}(c) = \mathsf{AV}(c^*) + 1 - \mathsf{AV}(c)$ be the minimum number of AppAdd operations needed to make $c$ have AV score at least $\mathsf{AV}(c^*) + 1$. We order the candidates in $C \setminus (J \cup C^>(c^*))$ due to a nondecreasing order of $\mathsf{diff}(c)$, and let $A$ be the set of the first $k - |C^>(c^*)|$ candidates in the order. If $\sum_{c \in A} \mathsf{diff}(c) \leq \ell$ we conclude that $I$ is a Yes-instance; otherwise, $I$ is a No-instance. □

An important base for the correctness of the algorithm in the proof of Theorem 4.2 is that adding candidates in a vote does not change the scores of other candidates, which allows us to solve the instance greedily. However, this is not the case in SAV and NSAV voting, where adding a candidate in a vote increases the score of this candidate but decreases the scores of other candidates in this vote. The difference of the behavior between AV and SAV/NSAV essentially distinguishes the complexity of the bribery problems under these rules.

Theorem 4.3. *DAppAddB-SAV and DAppAddB-NSAV are NP-hard even if $k = 1$.*

Proof. We give only the proof for SAV via a reduction from the RX3C problem. The reduction for NSAV is a modification of the reduction for SAV based on Lemma 4.1.

Let $(A, \mathcal{H})$ be an instance of RX3C where $|A| = |\mathcal{H}| = 3\kappa$. We assume that $\kappa > 4$ and $\kappa$ is even which does not change the complexity of the problem. We create an instance denoted by $((C, V), J \subseteq C, \ell, k)$ of DAppAddB-SAV as follows.

First, we create a set of $3\kappa$ candidates corresponding to $A$, one for each. Let $c(a)$ denote the candidate created for $a \in A$ and let $C(A) = \{c(a) : a \in A\}$. In addition, we create a candidate denoted by $p$. We define $C = C(A) \cup \{p\}$. We let $J = C(A)$, set $k = 1$ and $\ell = \kappa$. The multiset $V$ of votes comprises of the following votes. First, we create $\frac{3}{4}\kappa^2 - 3\kappa$ votes each of which approves all candidates except $p$. As we assumed that $\kappa > 4$ and $\kappa$ is even, $\frac{3}{4}\kappa^2 - 3\kappa$ is a positive integer. In addition to the above votes, for each $H \in \mathcal{H}$, we create one vote $v(H)$ which approves exactly the three candidates corresponding to its three elements, i.e., $v(H) = \{c(a) : a \in H\}$. This completes the construction. Observe that in this election the SAV score of the nondistinguished candidate $p$ is 0 and that of everyone else is $\left(\frac{3}{4}\kappa^2 - 3\kappa\right) \cdot \frac{1}{3\kappa} + 3 \times \frac{1}{3} = \frac{\kappa}{4}$. It remains to show the correctness of the reduction. Notice that as $C = J \cup \{p\}$ and $k = 1$, the question in consideration is equivalent to whether we can make at most $\ell = \kappa$ additions so that $\{p\}$ becomes the unique winning 1-committee.

($\Rightarrow$) Assume that $\mathcal{H}' \subseteq \mathcal{H}$ is an exact set cover of $A$. Consider the election after the following modifications: for each $H \in \mathcal{H}'$, add $p$ into the vote $v(H)$, i.e., reset $v(H) := v(H) \cup \{p\}$. As $|\mathcal{H}'| = \kappa$, we make exactly $\kappa$ additions. In this election, the votes approving $p$ are exactly those corresponding to $\mathcal{H}'$. As each of these votes approves four candidates now and there are exactly $\kappa$ of them, the SAV score of $p$ in this election is $\frac{\kappa}{4}$. For each candidate $c(a)$ where $a \in A$, its SAV score decreases when we add $p$ in some vote $v(H)$ such that $a \in H \in \mathcal{H}'$ by $\frac{1}{3} - \frac{1}{4} = \frac{1}{12}$. As $\mathcal{H}'$ is an exact 3-set cover, there is exactly one such vote. Therefore, after the above modifications, the SAV score of $c(a)$ where $a \in A$ decreases to $\frac{\kappa}{4} - \frac{1}{12}$, leading to $\{p\}$ being the unique winning 1-committee.

($\Leftarrow$) Assume that we can make at most $\ell = \kappa$ additions so that $\{p\}$ is the unique winning 1-committee. Without loss of generality, assume that exactly $t$ votes among the $\frac{3}{4}\kappa^2 - 3\kappa$ votes approving $J$ are modified. Observe that for these votes, we can only add $p$ to them. We claim first that $t = 0$ in fact. Assume, for the sake of contradiction that $t > 0$. Then, at most $\kappa - 1$ votes corresponding to $\mathcal{H}$ can be modified. This implies that there exists at least one distinguished candidate $c(a)$ where $a \in A$ such that none of the three votes $v(H)$ such that $a \in H \in \mathcal{H}$ is modified. This further means that after the modifications, the candidate $c(a)$ has SAV score

$$\left(\frac{3}{4}\kappa^2 - 3\kappa - t\right) \cdot \frac{1}{3\kappa} + \frac{t}{3\kappa + 1} + 1 = \frac{\kappa}{4} - \frac{t}{3\kappa} + \frac{t}{3\kappa + 1}.$$

However, after the modifications the SAV score of $p$ can be at most $\frac{t}{3\kappa+1} + \frac{\kappa - t}{4}$ which is strictly smaller than that of $c(a)$ for $\kappa > 4$. This contradicts that after the modifications, $\{p\}$ is the unique winning 1-committee, and our claim is proved.

Now we can assume that all modified votes are from those corresponding to $\mathcal{H}$. Moreover, we can observe that when some $v(H)$ where $H \in \mathcal{H}$, is supposed to be modified, it is optimal to add only $p$ in the vote. Therefore, under this claim, exactly $\kappa$ votes corresponding to $\mathcal{H}$ are modified and each of them is modified by adding $p$. In this case, the SAV score of $p$ is exactly $\frac{\kappa}{4}$ after the modifications. Because $\{p\}$ is the unique winning 1-committee after the modifications, it must be that for every candidate $c(a)$ where $a \in A$, at least one vote $v(H)$ such that $a \in H$ is modified so that the SAV score of $c(a)$ is decreased, implying that the subcollection corresponding to the set of modified votes is an exact set cover of $A$. □

## 4.2 Approval Deletion

This section explores the atomic operation AppDel. For AV, we can obtain a polynomial-time solvability result again.

Theorem 4.4. *DAppDelB-AV is polynomial-time solvable.*

Proof. Given an instance $((C, V), J \subseteq C, k, \ell)$ of DAppDelB-AV, we first check if the number of nondistinguished candidates who are approved by at least one vote is at most $k - 1$. If this is the case, we immediately conclude that the given instance is a No-instance. Otherwise, our algorithm proceeds by exhaustively applying the following reduction rule.

**Reduction Rule.** Let $c^* \in J$ be a candidate such that $\mathsf{AV}(c^*) \geq \mathsf{AV}(c)$ for all $c \in J$. If the number of nondistinguished candidates whose AV scores are at least $\mathsf{AV}(c^*) + 1$ is at most $k - 1$, remove $c^*$ from any arbitrary vote which approves $c^*$, and decrease $\ell$ by one.

After exhaustively applying the above reduction rule, we conclude that the given instance is a Yes-instance if and only if $\ell \geq 0$.

The polynomial-time solvability follows from that each application of the above reduction rule takes polynomial time, and we apply the rule at most $n \cdot m$ times, where $n$ and $m$ respectively denote the number of votes and the number of candidates. □

However, for SAV and NSAV, we again have hardness results.

THEOREM 4.5 (★). *DAppDelB-SAV and DAppDelB-NSAV are NP-hard even when* $k = 1$.

Without stopping here, we also show W[1]-hardness results for SAV and NSAV with respect to the parameters $\ell$ and $k$. This holds even when there is only one distinguished candidate.

THEOREM 4.6 (★). *DAppDelB-SAV and DAppDelB-NSAV are W[1]-hard with respect to both the parameter* $\ell$ *and the parameter* $k$. *Moreover, the results hold even when* $|J| = 1$.

## 4.3 Vote Change

From this section, we study vote-level operations. We show that problems associated with these operations are generally NP-hard or W[1]-hard even in some special cases, and this is already the case for AV.

THEOREM 4.7 (★). $r$*-DVCB-AV for all possible values of* $r \geq 4$ *are NP-hard even when* $k = 1$.

We point out that in the proof of Theorem 4.7, the number of distinguished candidates is again not bounded by a constant. One may wonder whether we have FPT algorithms with respect to the number of distinguished candidates, or the combined parameter of $k$ and $|J|$. The next result destroys this hope.

THEOREM 4.8. $r$*-DVCB-AV for all possible values of* $r \geq 3$ *is W[1]-hard with respect to both* $\ell$ *and* $k$. *This holds even when* $|J| = 1$.

PROOF. We prove the theorem via a reduction from the $\kappa$-CLIQUE problem restricted to regular graphs. Let $(G = (U, A), \kappa)$ be an instance of the $\kappa$-CLIQUE problem where every vertex in $G$ has degree exactly $d$ for some positive integer $d$. Without loss of generality, we assume that $\kappa \geq 2$. Moreover, we assume that $d > \kappa^3$ since otherwise the instance can be solved in FPT time with respect to $\kappa$. Let $r$ be an integer at least 3. We construct an $r$-DVCB-AV instance $((C, V), J \subseteq C, \ell, k)$ as follows. The candidate set is $C = U \cup \{p\}$ where $p \notin U$ is the only distinguished candidate, i.e., $J = \{p\}$. We create $m$ votes corresponding to the edges in $G$, where $m$ is the number of vertices in $G$. In particular, for each edge $\{u, u'\} \in A$, we create a vote approving all candidates except $u$ and $u'$. In addition, we create $d + 1 - \frac{(\kappa-1)\cdot(\kappa+2)}{2}$ votes approving all candidates except $p$. Note that under our assumption $d > \kappa^3$ and $\kappa \geq 2$, $d + 1 - \frac{(\kappa-1)\cdot(\kappa+2)}{2}$ is a positive integer. We complete the construction by setting $\ell = \frac{\kappa\cdot(\kappa-1)}{2}$ and $k = \kappa$, i.e., we are allowed to change at most $\frac{\kappa\cdot(\kappa-1)}{2}$ votes and we aim to select exactly $\kappa$ winners.

Now we show the correctness of the reduction. Let us first consider the AV scores of the candidates. The AV score of $p$ is $\mathsf{AV}(p) = m$, and the AV score of every other candidate $u \in U$ is

$$\begin{aligned}\mathsf{AV}(u) &= \left(d + 1 - \frac{(\kappa-1)\cdot(\kappa+2)}{2}\right) + (m - d)\\ &= 1 + m - \frac{(\kappa-1)\cdot(\kappa+2)}{2}.\end{aligned}$$

Under the assumption $\kappa \geq 2$, it holds that $\mathsf{AV}(u) < \mathsf{AV}(p)$ for every candidate $u \in U$.

($\Rightarrow$) Assume that there is a clique $U' \subseteq U$ of size $\kappa$ in the graph $G$. We modify all votes corresponding to edges between vertices in the clique $U'$, i.e., all edges in $G[U']$. Clearly, there are exactly $\frac{\kappa\cdot(\kappa-1)}{2} = \ell$ such votes. We modify them so that all candidates except $p$ are approved in these votes. The distance between each new vote and its original vote is exactly 3 which does not disobey the distance restriction. As a vote corresponding to some edge $\{u, u'\}$ initially approves $p$ but approves neither $u$ nor $u'$, after modifying this vote, the AV score of $p$ decreases by one and the AV scores of both $u$ and $v$ increase by one. So, after all these modifications the AV score of $p$ becomes $m - \frac{\kappa\cdot(\kappa-1)}{2}$. For each $u \in U'$, as there are exactly $\kappa - 1$ edges incident to $u$ in $G[U']$, after modifying the above $\ell$ votes, the AV score of $u$ becomes $\mathsf{AV}(u) + \kappa - 1 = 1 + m - \frac{\kappa\cdot(\kappa-1)}{2}$, which is strictly greater than the final score of $p$, implying that $p$ cannot be in any winning $k$-committees after all the modifications.

($\Leftarrow$) Assume that we can change at most $\ell = \frac{\kappa\cdot(\kappa-1)}{2}$ votes so that $p$ is not included in any winning $k$-committee. Observe that it is always better to modify votes corresponding to the edges than those who already disapprove $p$ and approve all the other candidates. Moreover, observe that it is optimal to full use the number of operations, and when a vote corresponding to an edge is determined to be modified, it is optimal to change it so that the vote approves all candidates except the distinguished candidate $p$. The final AV score of $p$ is then determined as $\mathsf{AV}(p) - \ell = m - \frac{\kappa\cdot(\kappa-1)}{2}$. This implies that after the changes, there must be at least $k$ candidates whose AV scores increase by at least $\kappa - 1$ each. Note that when a vote corresponding to some edge $\{u, u'\}$ is changed, only the AV scores of $u$ and $u'$ increase, each by one. Let $U'$ be the set of candidates whose scores are increased after the changes of the votes in a solution. Therefore, $U'$ consists of the vertices spanned by all edges whose corresponding votes are changed. Due to the above analysis, changing $\ell = \frac{\kappa\cdot(\kappa-1)}{2}$ edge-votes can increase the total AV scores of candidates in $U'$ by at most $\kappa \cdot (\kappa - 1)$. This directly implies that $U'$ consists of exactly $k$ candidates. From the fact that $\frac{\kappa\cdot(\kappa-1)}{2}$ edges span exactly $\kappa$ vertices if and only if the set of spanned vertices is a clique, we know that $U'$ is a clique in $G$. □

THEOREM 4.9. $r$*-DVCB-SAV and* $r$*-VC-NSAV for all integers* $r \geq 4$ *are NP-hard. This hold even if* $k = 1$.

Regarding fixed-parameter intractability results, we have the following theorem.

THEOREM 4.10 (★). $r$*-DVCB-SAV and* $r$*-DVCB-NSAV for all possible values of* $r \geq 1$ *are W[1]-hard with respect to both the parameter* $\ell$ *and the parameter* $k$. *The results hold even when* $|J| = 1$.

## 4.4 Vote Addition Change

In the previous section, we showed that $r$-DVCB-AV is already NP-hard when $k = 1$. However, this is not the case for vote-level addition operation.

Theorem 4.11. *$r$-DVACB-AV for all possible values of $r$ is polynomial-time solvable if $k = 1$.*

Proof. Let $((C, V), J \subseteq C, k = 1, \ell)$ be an instance of $r$-DVACB-AV, where $r$ is a nonnegative integer. If $r = 0$, we solve the instance by directly checking if none of the distinguished candidates is a winner. Now we consider the case where $r > 0$. Let $n = |V|$. In addition, let $s$ be the maximum AV score of the distinguished candidates, i.e., for all $c \in J$ it holds that $\mathsf{AV}(c) \leq s$ and there is at least one candidate $c' \in J$ such that $\mathsf{AV}(c') = s$. Finally, let $c^* \in C \setminus J$ be a candidate with the maximum AV score among all those in $C \setminus J$. Our algorithm goes as follows. If $c^*$ already has AV score at least $s+1$, we return "Yes". Otherwise, if $\mathsf{AV}(c^*) + \min\{\ell, n - |V(c^*)|\} \geq s + 1$, we return "Yes" too. The reason is that in this case we can select arbitrarily $\min\{\ell, n\}$ votes in $V \setminus V(c^*)$ and add $c^*$ into each of the selected votes so that the AV score of $c^*$ is $s+1$. If none of the above two cases occurs, we return "No". □

However, we have fixed-parameter intractable result when $k$ and $\ell$ are parameters even when other parameters are constants.

Theorem 4.12 (★). *$r$-DVACB-AV for all integers $r \geq 2$ is W[1]-hard with respect to the parameters $\ell$ and $k$. Moreover, this holds even when $|J| = 1$.*

Unlike AV, for SAV and NSAV, we already have NP-hardness even when both $k$ and $r$ are equal to 1.

Theorem 4.13 (★). *$r$-DVACB-SAV and $r$-DVACB-NSAV for all integers $r \geq 1$ are NP-hard. Moreover, this holds even when $k = 1$.*

## 4.5 Vote Deletion Change

For the vote-level deletion operation, we have an NP-hardness result for AV even when we want to elect only one winner.

Theorem 4.14 (★). *$r$-DVDCB-AV is NP-hard for all possible integers $r \geq 3$ even when $k = 1$.*

Next, we show that if every vote is only allowed to delete at most one approved candidate, the problem becomes polynomial-time solvable, regardless of the values of $k$.

Theorem 4.15. *$r$-DVDCB-AV is polynomial-time solvable if $r = 1$.*

Proof. We solve the problem by reducing it to the maximum matching problem which is polynomial-time solvable [19, 25]. Let $((C, V), J \subseteq C, k, \ell)$ be an instance of $r$-DVDCB-AV. We calculate the AV scores of all nondistinguished candidates and order them according to their scores from the highest to the lowest with ties being broken arbitrarily. Let $s$ denote the AV score of the $k$-th candidate in the order. Our goal is to select at most $\ell$ votes and remove some distinguished candidates from these votes, one from each, so that the AV score of every distinguished candidate is at most $s-1$. Let $J'$ be the set of distinguished candidates who have AV scores at least $s$. We create a bipartite graph. Particularly, we create a vertex for each distinguished candidate in $J'$, and create a vertex for each vote which approves at least one candidate in $J'$. We connect a vote-vertex with a candidate-vertex if and only if the vote approves this candidate. Then, we do the following. For each distinguished candidate $c \in J'$ of score $\mathsf{AV}(c) \geq s + 1$, we create $\mathsf{AV}(c) - s$ copies of the corresponding vertex (each copy also has the same neighbors as the original vertex). Note that we can immediately conclude that the given instance is a No-instance if $\sum_{c \in J'} (\mathsf{AV}(c) - s + 1) > \ell$. So, let us assume that this is not the case. Then, we calculate a maximum matching. If all candidate-vertices and all of their copies are saturated by the matching, we conclude that the instance is a Yes-instance; otherwise, the instance is a No-instance. □

Theorem 4.16 (★). *$r$-DVDCB-SAV and $r$-DVDCB-NSAV are NP-hard for all possible integers $r \geq 3$ even when $k = 1$ and every voter approves at most three candidates.*

Next, we show that for SAV and NSAV, destructive bribery with the vote deletion operation is W[1]-hard.

Theorem 4.17 (★). *$r$-DVDCB-SAV and $r$-DVDCB-NSAV for all possible values of $r \geq 1$ are W[1]-hard with respect to both $\ell$ and $k$. This holds even when $|J| = 1$.*

# 5 FIXED-PARAMETER TRACTABILITY

In the previous sections, we have obtained many intractability results and a few polynomial-time solvability results in some special cases. This section aims to explore fixed-parameter tractable algorithms with respect to three natural parameters: the number of candidates $m$, the number of voters $n$, and the number of distinguished candidates $|J|$. As $|J| \leq m$ in each problem instance studied in this paper, any FPT-algorithm with respect to $|J|$ carries over to $m$ directly. These three parameters are relevant to many real-world applications and have received extensive study [3, 11, 15, 40, 47].

We have shown that $r$-VC-AV and $r$-VAC-AV are NP-hard even when there is only one distinguished candidate but left whether $r$-VDC-AV is also NP-hard in this case unexplored in the previous sections. We answer this question now. Interestingly, we show that $r$-VDC-AV is FPT with respect to $|J|$, standing in a sharp contrast to the NP-hardness of the other two problems even when $|J| = 1$.

Theorem 5.1. *$r$-DVDCB-AV for all possible values of $r$ is FPT with respect to the number of distinguished candidates.*

Proof. Let $((C, V), J \subseteq C, k, \ell)$ be an instance of $r$-DVDCB-AV where $r$ is a nonnegative integer. Our FPT-algorithm is based on integer-linear programming (ILP) formulation. For each subset $A \subseteq J$, let $V(A) = \{v \in V : v \cap J = A\}$ denote the multiset of votes approving exactly the candidates in $A$ among all distinguished candidates. Let $n(A) = |V(A)|$. We calculate the AV scores of all nondistinguished candidates and rank them according to their AV scores, from those with the highest score to those with the lowest score, with ties being broken arbitrarily. Let $s$ denote the AV score of the $k$-th candidate in this rank. For each $A \subseteq J$ and each $B \subseteq A$ such that $|B| \leq r$, we create a nonnegative integer variable $x_{A,B}$ which indicates that in a solution we change $x_{A,B}$ votes in $V(A)$ so that exactly the candidates in $B$ are removed from these votes. We have at most $4^{|J|}$ variables. The constraints are as follows. As we change at most $\ell$ votes, it holds that $\sum_{B \subseteq A \subseteq J, |B| \leq r} x_{A,B} \leq \ell$. For each $A \subseteq J$, it holds that $\sum_{B \subseteq A, |B| \leq r} x_{A,B} \leq n(A)$. Finally, to ensure that the final score of every distinguished candidate is at most $s-1$, for every $c \in J$, it holds that $\mathsf{AV}_{(C,V)}(c) - \sum_{c\in B, B\subseteq A\subseteq J, |B|\leq r} x_{A,B} \leq s-1$. This ILP can be solved in FPT time with respect to $|J|$ due to the Lenstra's theorem [30]. □

Note that the fixed-parameter tractability with respect to $|J|$ does not hold for SAV and NSAV as we have shown in the previous sections that $r$-DVDCB-SAV and $r$-DVDCB-NSAV are W[1]-hard with respect to $\ell$ and $k$ even when $|J| = 1$. This is essentially because that in AV, deleting a candidate from a vote does not affect the AV scores of other candidates in the vote, but in SAV and NSAV, deleting a candidate from a vote increases the SAV and NSAV scores of other candidates in the vote. The behavior of AV allows us to only focus on removing distinguished candidates but the behavior of SAV and NSAV asks us to pay attention to all candidates.

Based on ILP formulations again, we can show that all problems studied in this paper are FPT with respect to a larger parameter, namely, the number of candidates $m$. This holds for all five rules studied in the paper. At a high level, our algorithm first guesses the exact winning $k$-committees, each of which does not include any distinguished candidates. There are at most $2^{\binom{m}{k}} \leq 2^{2^m}$ guesses and each guess involves at most $2^m$ committees. For each guessed class of winning $k$-committees, we provide an ILP formulation. Particularly, we partition the votes into at most $2^m$ groups, each group consists of all votes approving the same set of candidates. Then, for each group, we introduce $2^m$ nonnegative integer variables, each of which corresponds to a subset $C'$ of candidates and indicates how many votes from the group are changed into votes approving exactly the candidates in $C'$. The constraints are derived to ensure that all $k$-committees in the guessed class have the same score which is strictly higher than that of any $k$-committees not in the class. For vote-level operations with distance bound $r$, we should also constraint the variables corresponding to a group of votes and a subset $C'$ to be 0 if the distance between each vote in the group and $C'$ is larger than $r$. The FPT-running time follows from that we need to solve at most $2^{2^m}$ ILPs each of which has at most $2^m \cdot 2^m = 4^m$ variables, and ILP is FPT with respect to the number of variables [30].

Theorem 5.2 (★). *For $X \in \{AV, SAV, NSAV, CCAV, PAV\}$, the problems DAppAddB-X, DAppDelB-X, $r$-DVCB-X, $r$-DVACB-X, and $r$-DVDCB-X are FPT with respect to the number of candidates. The results for $r$-DVCB-X, $r$-DVACB-X, and $r$-DVDCB-X hold for all possible values of $r$.*

Finally, we study the parameter $n$, the number of votes.

Theorem 5.3. *$r$-DVDCB-AV for all possible values of $r$ can be solved in $O^*(2^n)$ time, where $n$ denotes the number of votes.*

Proof. Let $((C, V), J \subseteq C, k, \ell)$ be an $r$-DVDCB-AV instance where $r$ is an integer. We order all nondistinguished candidates according to their AV scores, from the highest to the lowest, with ties being broken arbitrarily. Let $s$ denote the score of the $k$-th candidate in the order. Let $J' = \{c \in J : \mathsf{AV}(c) \geq s\}$ be the set of distinguished candidates whose AV scores are at least $s$. We guess the $\ell$ votes which need to be modified to make all distinguished candidates be excluded from all winning $k$-committees. Precisely, we split the instance into at most $2^n$ subinstances each of which takes as input the original instance together with a subset $V' \subseteq V$ of $\ell$ votes, and the question is whether we can modify exactly the votes in $V'$ to exclude all distinguished candidates from any winning $k$-committee. Clearly, the original instance is a Yes-instance if and only if at least one of the subinstances is a Yes-instance. To solve a subinstance corresponding to a subset $V' \subseteq V$, we reduce it to a maximum network flow instance. Particularly, in the maximum network flow instance, we have a source node $v^+$ and a sink node $v^-$. Moreover, for each vote $v \in V'$, we create a node denoted still by $v$ for simplicity. For each distinguished candidate $c \in J'$ we create a node denoted still by $c$ for simplicity. The arcs are as follows. There is an arc from the source node $v^+$ to every vote-vertex $v$ with capacity $\min\{r, |v \cap J'|\}$. For a vote-node $v$ and a candidate-node $c$, there is an arc from $v$ to $c$ with capacity 1 if and only if $v$ approves $c$, i.e., $c \in v$. Finally, for each candidate-node $c \in J'$ there is an arc from $c$ to the sink node $v^-$ with capacity $\mathsf{AV}(c) - s + 1$. It is easy to see that the subinstance is a Yes-instance if and only if the above constructed network has a flow of size $\sum_{c\in J'}(\mathsf{AV}(c) - s + 1)$. The theorem follows from that the maximum network flow problem can be solved in polynomial-time (see, e.g., [32]). □

We can show that for the other two vote-level operations, the corresponding problems are also FPT with respect to $n$ when $r = m$. Precisely, we enumerate all subsets of at most $\ell$ votes which are considered to be modified. Once the modified votes are determined, we can solve the instance greedily when the distance restriction is dropped: for the VAC operation, we add all nondistinguished candidates in these votes, and for the VC operation, we let these votes approve exactly all nondistinguished candidates.

Theorem 5.4 (★). *$r$-DVCB-AV and $r$-DVACB-AV for $r = m$ can be solved in $O^*(2^n)$ time where $n$ is the number of votes.*

## 6 CONCLUSION

We have studied the (parameterized) complexity of five destructive bribery problems in the setting of approval-based multi-winner voting. These problems model the scenario where a briber aims to exclude all of a given set of candidates from having any chance to win by bribing some voters without exceeding her/his budget. Our study significantly complements previous study because bribery problems for ABM rules in the destructive model have not been widely studied prior to our work. For the five well-studied ABM rules AV, SAV, NSAV, CCAV, and PAV, we provided a comprehensive landscape of the (parameterized) complexity of these problems. Our results are summarized in Table 2.

There are many possibilities for future work. First, one can always start by resolving open problems left. For instance, all problems considered in this paper are FPT with respect to the number of candidates. However, we have only a few FPT-algorithms with respect to the number of votes, leaving many cases remained open. It should be noted that many control and manipulation problems are already NP-hard even when the number of voters is a constant [3, 8, 15]. Second, our study is purely theoretical analysis. One can conduct experimental work to investigate whether these problems are really hard to solve in practice. Third, the complexity of these problems in special dichotomous domains is widely open. For concepts of dichotomous domains we refer to [20, 21, 42].

## REFERENCES


[1] H. Aziz, M. Brill, V. Conitzer, E. Elkind, R. Freeman, and T. Walsh. 2017. Justified Representation in Approval-Based Committee Voting. *Soc. Choice Welfare* 48, 2 (2017), 461–485.
[2] H. Aziz, S. Gaspers, J. Gudmundsson, S. Mackenzie, N. Mattei, and T. Walsh. 2015. Computational Aspects of Multi-winner Approval Voting. In *AAMAS*. 107–115.
[3] G. Bachmeier, F. Brandt, C. Geist, P. Harrenstein, K. Kardel, D. Peters, and H. G. Seedig. 2019. $k$-Majority Digraphs and the Hardness of Voting with a Constant Number of Voters. *J. Comput. Syst. Sci.* 105 (2019), 130–157.
[4] J. J. Bartholdi III and J. B. Orlin. 1991. Single Transferable Vote Resists Strategic Voting. *Soc. Choice Welfare* 8, 4 (1991), 341–354.
[5] J. J. Bartholdi III, C. A. Tovey, and M. A. Trick. 1989. The Computational Difficulty of Manipulating an Election. *Soc. Choice Welfare* 6, 3 (1989), 227–241.
[6] J. J. Bartholdi III, C. A. Tovey, and M. A. Trick. 1992. How Hard Is It to Control an Election? *Math. Comput. Model.* 16, 8-9 (1992), 27–40.
[7] D. Baumeister, T. Hogrebe, and L. Rey. 2019. Generalized Distance Bribery. In *AAAI*. 1764–1771.
[8] N. Betzler, R. Niedermeier, and G. J. Woeginger. 2011. Unweighted Coalitional Manipulation under the Borda Rule Is NP-Hard. In *IJCAI*. 55–60.
[9] N. Betzler, A. Slinko, and J. Uhlmann. 2013. On the Computation of Fully Proportional Representation. *J. Artif. Intell. Res.* 47 (2013), 475–519.
[10] S. J. Brams and D. M. Kilgour. 2014. Satisfaction Approval Voting. In *Voting Power and Procedures*, R. Fara, D. Leech, and M. Salles (Eds.). Springer International Publishing, 323–346.
[11] R. Bredereck, P. Faliszewski, R. Niedermeier, P. Skowron, and N. Talmon. 2015. Elections with Few Candidates: Prices, Weights, and Covering Problems. In *ADT*. 414–431.
[12] R. Bredereck, P. Faliszewski, R. Niedermeier, and N. Talmon. 2016. Complexity of Shift Bribery in Committee Elections. In *AAAI*. 2452–2458.
[13] L. Cai. 2008. Parameterized Complexity of Cardinality Constrained Optimization Problems. *Comput. J.* 51, 1 (2008), 102–121.
[14] J. R. Chamberlin and P. N. Courant. 1983. Representative Deliberations and Representative Decisions: Proportional Representation and the Borda Rule. *Am. Polit. Sci. Rev.* 77, 3 (1983), 718–733.
[15] J. Chen, P. Faliszewski, R. Niedermeier, and N. Talmon. 2017. Elections with Few Voters: Candidate Control Can Be Easy. *J. Artif. Intell. Res.* 60 (2017), 937–1002.
[16] P. Dey. 2019. Local Distance Restricted Bribery in Voting. In *AAMAS*. 1925–1927.
[17] R. Downey. 2012. A Parameterized Complexity Tutorial. In *LATA*. 38–56.
[18] R. G. Downey and M. R. Fellows. 1992. Fixed-Parameter Intractability. In *Structure in Complexity Theory Conference*. 36–49.
[19] J. Edmonds. 1965. Paths, Trees, and Flowers. *Can. J. of Math* 17 (1965), 449–467.
[20] E. Elkind and M. Lackner. 2015. Structure in Dichotomous Preferences. In *IJCAI*. 2019–2025.
[21] E. Elkind, M. Lackner, and D. Peters. 2017. Structured Preferences. In *Trends in Computational Social Choice*. AI Access, Chapter 10, 187–207.
[22] G. Erdélyi, C. Reger, and Y. Yang. 2020. The Complexity of Bribery and Control in Group Identification. *Auton. Agent Multi-Ag.* 34, 1 (2020). Article No. 8.
[23] P. Faliszewski, P. Skowron, and N. Talmon. 2017. Bribery as a Measure of Candidate Success: Complexity Results for Approval-Based Multiwinner Rules. In *AAMAS*. 6–14.
[24] L. Fernández and J. A. Fisteus. 2019. Monotonicity Axioms in Approval-based Multi-winner Voting Rules. In *AAMAS*. 485–493.
[25] Z. Galil. 1986. Efficient Algorithms for Finding Maximum Matching in Graphs. *ACM Comput. Surv.* 18, 1 (1986), 23–38.
[26] G. Gawron and P. Faliszewski. 2019. Robustness of Approval-Based Multiwinner Voting Rules. In *ADT*. 17–31.
[27] T. F. Gonzalez. 1985. Clustering to Minimize the Maximum Intercluster Distance. *Theor. Comput. Sci.* 38 (1985), 293–306.
[28] D. M. Kilgour and E. Marshall. 2012. Approval Balloting for Fixed-Size Committees. In *Electoral Systems*. Springer Berlin Heidelberg, 305–326.
[29] R. LeGrand. 2004. *Analysis of the Minimax Procedure*. Technical Report. Department of Computer Science and Engineering, Washington University, St. Louis, Missouri.
[30] H. W. Lenstra. 1983. Integer Programming with a Fixed Number of Variables. *Math. Oper. Res.* 8, 4 (1983), 538–548.
[31] R. Meir, A. D. Procaccia, J. S. Rosenschein, and A. Zohar. 2008. Complexity of Strategic Behavior in Multi-winner Elections. *J. Artif. Intell. Res.* 33 (2008), 149–178.
[32] J. B. Orlin. 2013. Max Flows in $O(nm)$ Time, or Better. In *STOC*. 765–774.
[33] D. Peters. 2018. Single-Peakedness and Total Unimodularity: New Polynomial-Time Algorithms for Multi-winner Elections. In *AAAI*. 1169–1176.
[34] L. Sánchez-Fernández, E. Elkind, M. Lackner, N. Fernández, J. A. Fisteus, P. B. Val, and P. Skowron. 2017. Proportional Justified Representation. In *AAAI*. 670–676.
[35] P. Skowron and P. Faliszewski. 2017. Chamberlin-Courant Rule with Approval Ballots: Approximating the MaxCover Problem with Bounded Frequencies in FPT Time. *J. Artif. Intell. Res.* 60 (2017), 687–716.
[36] P. Skowron, P. Faliszewski, and A. M. Slinko. 2015. Achieving Fully Proportional Representation: Approximability Results. *Artif. Intell.* 222 (2015), 67–103.
[37] T. N. Thiele. 1895. Om flerfoldsvalg. *Oversigt over det Kongelige Danske Videnskabernes Selskabs Forhandlinger* (1895), 415–441.
[38] C. A. Tovey. 2002. Tutorial on Computational Complexity. *Interfaces* 32, 3 (2002), 30–61.
[39] D. B. West. 2000. *Introduction to Graph Theory*. Prentice-Hall.
[40] Y. Yang. 2014. Election Attacks with Few Candidates. In *ECAI*. 1131–1132.
[41] Y. Yang. 2019. Complexity of Manipulating and Controlling Approval-Based Multiwinner Voting. In *IJCAI*. 637–643.
[42] Y. Yang. 2019. On the Tree Representations of Dichotomous Preferences. In *IJCAI*. 644–650.
[43] Y. Yang and D. Dimitrov. 2018. How Hard Is It to Control a Group? *Auton. Agent Multi-Ag.* 32, 5 (2018), 672–692.
[44] Y. Yang, Y. R. Shrestha, and J. Guo. 2019. On the Complexity of Bribery with Distance Restrictions. *Theor. Comput. Sci.* 760 (2019), 55–71.
[45] Y. Yang and J. Wang. 2017. Anyone But Them: The Complexity Challenge for a Resolute Election Controller. In *AAMAS*. 1133–1141.
[46] Y. Yang and J. Wang. 2018. Parameterized Complexity of Multi-winner Determination: More Effort Towards Fixed-Parameter Tractability. In *AAMAS*. 2142–2144.
[47] A. Zhou, Y. Yang, and J. Guo. 2019. Parameterized Complexity of Committee Elections with Dichotomous and Trichotomous Votes. In *AAMAS*. 503–510.