Bribery Can Get Harder in Structured Multiwinner Approval Election

Bartosz Kusek
AGH University of Science and Technology
Kraków, Poland
kussy.kusy@gmail.com

Robert Bredereck
Humboldt-Universität zu Berlin, TU Clausthal
Berlin, Clausthal-Zellerfeld, Germany
robert.bredereck@tu-clausthal.de

Piotr Faliszewski
AGH University of Science and Technology
Kraków, Poland
faliszew@agh.edu.pl

Andrzej Kaczmarczyk
AGH University of Science and Technology
Kraków, Poland
andrzej.kaczmarczyk@agh.edu.pl

Dušan Knop
Czech Technical University in Prague
Prague, Czech Republic
dusan.knop@fit.cvut.cz

ABSTRACT

We study the complexity of constructive bribery in the context of structured multiwinner approval elections. Given such an election, we ask whether a certain candidate can join the winning committee by adding, deleting, or swapping approvals, where each such action comes at a cost and we are limited by a budget. We assume our elections to either have the candidate interval or the voter interval property, and we require the property to hold also after the bribery. While structured elections usually make manipulative attacks significantly easier, our work also shows examples of the opposite behavior. We conclude by presenting preliminary insights regarding the destructive variant of our problem.

KEYWORDS

bribery; structured domain; approval elections; complexity reversal

ACM Reference Format:
Bartosz Kusek, Robert Bredereck, Piotr Faliszewski, Andrzej Kaczmarczyk, and Dušan Knop. 2023. Bribery Can Get Harder in Structured Multiwinner Approval Election. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 9 pages.

1 INTRODUCTION

We study the complexity of bribery under the multiwinner approval rule, in the case where the voters' preferences are structured. Specifically, we use the bribery model of Faliszewski, Skowron, and Talmon [16], where one can either add, delete, or swap approvals, and we consider *candidate interval* and *voter interval* preferences [10].

In multiwinner elections, the voters express their preferences over the available candidates and use this information to select a winning committee (i.e., a fixed-size subset of candidates). We focus on one of the simplest and most common scenarios, where each voter specifies the approved candidates, and those with the highest numbers of approvals form the committee. Such elections are used, e.g., to choose city councils, boards of trustees, or to shortlist job candidates. Naturally, there are many other rules and scenarios, but they do not appear in practice as often as this simplest one. For more details on multiwinner voting, we point the readers to the overviews of Faliszewski et al. [15] and Lackner and Skowron [20].

In our scenario, we are given an election, including the contents of all the votes, and, depending on the variant, we can either add, delete, or swap approvals, but each such action comes at a cost. Our goal is to find a cheapest set of actions that ensure that a given candidate joins the winning committee. Such problems, where we modify the votes to ensure a certain outcome, are known under the umbrella name of *bribery*, and were first studied by Faliszewski, Hemaspaandra and Hemaspaandra [12], whereas our specific variant is due to Faliszewski, Skowron, and Talmon [16]. Historically, bribery problems indeed aimed to model vote buying, but currently more benign interpretations prevail. For example, Faliszewski, Skowron, and Talmon [16] suggest using the cost of bribery as a measure of a candidate's success: A candidate who did not win, but can be put into the committee at a low cost, certainly did better than one whose bribery is expensive. In particular, since our problem is used for post-election analysis, it is natural to assume that we know all the votes. For other similar interpretations, we point, e.g., to the works of Xia [30], Shiryaev, Yu, and Elkind [28], Bredereck et al. [7], Boehmer et al. [4], or Baumeister and Hogrebe [1]. Faliszewski and Rothe [14] give a more general overview of bribery problems.

We assume that our elections either satisfy the candidate interval (CI) or the voter interval (VI) property [10], which correspond to the classic notions of single-peakedness [3] and single-crossingness [23, 26] from the world of ordinal elections. Briefly put, the CI property means that the candidates are ordered and each voter approves some interval of them, whereas the VI property requires that the voters are ordered and each candidate is approved by some interval of voters. For example, the CI assumption can be used to model political elections, where the candidates appear on the left-to-right spectrum of opinions and the voters approve those, whose opinions are close enough to their own. Importantly, we require our elections to have the CI/VI property also after the bribery; this approach is standard in bribery problems with structured elections [6, 9, 22], as well as in other problems related to manipulating election results [13, 17, 29] (these references are examples only).

Example 1. *Let us consider a hotel in a holiday resort. The hotel has its base staff, but each month it also hires some additional help.*

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom.

For the coming month, the expectation is to hire extra staff for k days. Naturally, they would be hired for the days when the hotel is most busy (the decision to request additional help is made a day ahead, based on the observed load; the k days do not need to be consecutive). Since hotel bookings are typically made in advance, one knows which days are expected to be most busy. However, some people will extend their stays, some will leave early, and some will have to shift their stays. Thus the hotel managers would like to know which days are likely to become the busiest ones after such changes: Then they could inform the extra staff as to when they are expected to be needed, and what changes in this preliminary schedule might happen. Our bribery problem (for the CI setting) captures exactly the problem that the managers want to solve: The days are the candidates, k is the committee size, and the bookings are the approval votes (note that each booking must regard a consecutive set of days). Prices of adding, deleting, and moving approvals correspond to the likelihood that a particular change actually happens (the managers usually know which changes are more or less likely). Since the bookings must be consecutive, the election has to have the CI property also after the bribery. The managers can solve such bribery problem for each of the days and see which ones can most easily be among the k busiest ones.

Note that the value of k can be estimated from previous experience, be limited by the budget for the salaries for the additional workers, depend on the predicted workload, be subject to the availability of the additional workers, and the like. The managers may even want to solve several instances of the problem, with different values of k.

Example 2. *For the VI setting, let us consider a related scenario. There is a team of archaeologists who booked a set of excavation sites, each for some consecutive number of days (they work on several sites in parallel). The team may want to add some extra staff to those sites that require most working days. However, as in the previous example, the bookings might get extended or shortened. The team's manager may use bribery to evaluate how likely it is that each of the sites becomes one of the most work-demanding ones. In this case, the days are the voters, and the sites are the candidates.*

There are two main reasons why structured elections are studied. Foremost, as in the above examples, sometimes they simply capture the exact problem at hand. Second, many problems that are intractable in general, become polynomial-time solvable if the elections are structured. Indeed, this is the case for many NP-hard winner-determination problems [2, 10, 24] and for various problems where the goal is to make some candidate a winner [13, 21], including some bribery problems [6, 9]. There are also some problems that stay intractable even for structured elections [27, 31][1] as well as examples of complexity reversals, where assuming structured preferences turns a polynomial-time solvable problem into an intractable one. However, such reversals are rare and, to the best of our knowledge, so far were only observed by Menon and Larson [22], for the case of weighted elections with three candidates (but see also the work of Fitzsimmons and Hemaspaandra [17], who find complexity reversals that stem from replacing total ordinal votes with ones that include ties).

Our Contribution. We provide an almost complete picture of the complexity of bribery by either adding, deleting, or swapping approvals under the multiwinner approval voting rule, for the case of CI and VI elections, assuming either that each bribery action has identical unit price or that they can be priced individually (see Table 1). By comparing our results to those for the unrestricted setting, provided by Faliszewski, Skowron, and Talmon [16], we find that any combination of tractability and intractability in the structured and unrestricted setting is possible. For example:

1. Bribery by adding approvals is solvable in polynomial time irrespective if the elections are unrestricted or have the CI or VI properties.
2. Bribery by deleting approvals (where each deleting action is individually priced) is solvable in polynomial time in the unrestricted setting, but becomes NP-hard for CI elections (for VI ones it is still in P).
3. Bribery by swapping approvals only to the designated candidate (with individually priced actions) is NP-hard in the unrestricted setting, but becomes polynomial-time solvable both for CI and VI elections.
4. Bribery by swapping approvals (where each action is individually priced and we are not required to swap approvals to the designated candidate only) is NP-hard in each of the considered settings.

We largely focus on the constructive setting, where the goal is to ensure that some candidate belongs to at least one winning committee (indeed, all the results above are for this setting). However, we also give a glimpse of what happens in the destructive setting, where we want to ensure that a given candidate does not belong to any winning committees. In a certain sense, in this case we also observe a form of "reversal." Typically, destructive variants of bribery (and related) problems are at least as easy to work with as their constructive counterparts, and lead to more positive results. In our case—albeit this is mostly an intuitive feeling—the situation for CI elections is the opposite. Obtaining the results for the destructive setting seems more challenging and leads to less satisfying theorem statements (e.g., we need less appealing prices) than in the constructive setting.

We omit some proofs due to limited space, but include them in the full version of the paper (https://arxiv.org/abs/2209.00368).

Possibility of Complexity Reversals. So far, most of the problems studied for structured elections were subproblems of the unrestricted ones. For example, a winner determination algorithm that works for all elections, clearly also works for the structured ones and complexity reversal is impossible. The case of bribery is different because, by assuming structured elections, not only do we restrict the set of possible inputs, but also we constrain the possible actions. Yet, scenarios where bribery is tractable are rare, and only a handful of papers considered bribery in structured domains (we mention those of Brandt et al. [6], Fitzsimmons and Hemaspaandra [17], Menon and Larson [22], Elkind et al. [9]), so opportunities for observing complexity reversals were, so far, very limited. We show several such reversals, obtained for very natural settings.

2 PRELIMINARIES

For a positive integer t, we write $[t]$ to mean the set $\{1, \dots, t\}$. By writing $[t]_0$ we mean the set $[t] \cup \{0\}$.

[1]These references are not complete and are meant as examples.

	Unrestricted		Candidate Interval (CI)		Voter Interval (VI)	
(prices)	(unit)	(any)	(unit)	(any)	(unit)	(any)
AddApprovals	P	P Faliszewski et al. [2017a]	P [Thm. 2]	P [Thm. 2]	P [Thm. 1]	P [Thm. 1]
DelApprovals	P	P Faliszewski et al. [2017a]	?	NP-com. [Thm. 4]	P [Thm. 3]	P [Thm. 3]
SwapApprovals to p	P	NP-com. Faliszewski et al. [2017a]	P [Thm. 5]	P [Thm. 5]	P [Thm. 6]	P [Thm. 6]
SwapApprovals	P	NP-com. Faliszewski et al. [2017a]	NP-com. [Thm. 7]	NP-com. [Thm. 7]	?	NP-com. [Thm. 8]

Table 1: Our results for the CI and VI domains, together with those of Faliszewski, Skowron, and Talmon [16] for the unrestricted setting. SwapApprovals to p refers to the problem where each action has to move an approval to the preferred candidate.

Approval Elections. An *approval election* $E = (C, V)$ consists of a set of candidates $C = \{c_1, \dots, c_m\}$ and a collection of voters $V = \{v_1, \dots, v_n\}$. Each voter $v_i \in V$ has *an approval ballot* (or, equivalently, *an approval set*) which contains the candidates that v_i approves. We write v_i to refer both to the voter and to his or her approval ballot; the meaning will always be clear from the context.

A *multiwinner voting rule* is a function f that given an election $E = (C, V)$ and a committee size $k \in [|C|]$ outputs a nonempty family of winning committees (where each committee is a size-k subset of C). We disregard the issue of tie-breaking and assume all winning committees to be equally worthy, i.e., we adopt the nonunique winner model.

Given an election $E = (C, V)$, we let the approval score of a candidate $c \in C$ be the number of voters that approve c, and we denote it as $\mathrm{score}_E(c)$. The approval score of a committee $S \subseteq C$ is $\mathrm{score}_E(S) = \sum_{c \in S} \mathrm{score}_E(c)$. Given an election E and a committee size k, the *multiwinner approval voting* rule, denoted AV, outputs all size-k committees with the highest approval score. Occasionally we also consider the *single-winner approval rule*, which is defined in the same way as its multiwinner variant, except that the committee size is fixed to be one. For simplicity, in this case we assume that the rule returns a set of tied winners (rather than a set of tied size-1 winning committees).

Structured Elections. We focus on elections where the approval ballots satisfy either the *candidate interval (CI)* or the *voter interval (VI)* properties [10]:

(1) An election has the CI property (is a CI election) if there is an ordering of the candidates (called the *societal axis*) such that each approval ballot forms an interval with respect to this ordering.
(2) An election has the VI property (is a VI election) if there is an ordering of the voters such that each candidate is approved by an interval of the voters (for this ordering).

Given a CI election, we say that the voters have CI ballots or, equivalently, CI preferences; we use analogous conventions for the VI case. As observed by Elkind and Lackner [10], there are polynomial-time algorithms that test if a given election is CI or VI and, if so, provide appropriate orders of the candidates or voters; these algorithms are based on solving the *consecutive ones* problem [5].

Notation for CI Elections. Let us consider a candidate set $C = \{c_1, \dots, c_m\}$ and a societal axis $\rhd = c_1 c_2 \cdots c_m$. Given two candidates c_i, c_j, where $i \leq j$, we write $[c_i, c_j]$ to denote the approval set $\{c_i, c_{i+1}, \dots, c_j\}$.

Bribery Problems. We focus on the variants of bribery in multiwinner approval elections defined by Faliszewski, Skowron, and Talmon [16]. Let f be a multiwinner voting rule and let Op be one of AddApprovals, DelApprovals, and SwapApprovals operations (in our case f will either be AV or its single-winner variant). In the f-Op-Bribery problem we are given an election $E = (C, V)$, a committee size k, a preferred candidate p, and a nonnegative integer B (the budget). We ask if it is possible to perform at most B unit operations of type Op, so that p belongs to at least one winning committee (this is the constructive variant of the problem; requiring only minor adaptions in proofs, all our results for this variant also hold if we require that p belongs to all winning committees):

(1) For AddApprovals, a unit operation adds a given candidate to a given voter's ballot.
(2) For DelApprovals, a unit operation removes a given candidate from a given voter's ballot.
(3) For SwapApprovals, a unit operation replaces a given candidate with another one in a given voter's ballot.

Like Faliszewski, Skowron, and Talmon [16], we also study the variants of AddApprovals and SwapApprovals problems where each unit operation must involve the preferred candidate.

We are also interested in the priced variants of the above problems, where each unit operation comes at a cost that may depend both on the voter and the particular affected candidates; we ask if we can achieve our goal by performing operations of total cost at most B. We distinguish the priced variants by putting a dollar sign in front of the operation type. For example, $AddApprovals means a variant where adding each candidate to each approval ballot has an individual cost.

Bribery in Structured Elections. We focus on the bribery problems where the elections have either the CI or the VI property. For example, in the AV-$AddApprovals-CI-Bribery problem the input election has the CI property (under a given societal axis) and we ask if it is possible to add approvals with up to a given cost so that (a) the resulting election has the CI property for the same societal axis, and (b) the preferred candidate belongs to at least one winning committee. The VI variants are defined analogously (in particular, the voters' order witnessing the VI property is given and the election must still have the VI property with respect to this order after the bribery). The convention that the election must have the same structural property before and after the bribery, and the fact that the order witnessing this property is part of the input, is standard in the literature; see, e.g., the works of Faliszewski et al. [13], Brandt et al. [6], Menon and Larson [22], and Elkind et al. [9]. Further, it also follows naturally from some applications (as in the scenarios from Examples 1 and 2).

Computational Problems. For a graph G, by $V(G)$ we mean its set of vertices and by $E(G)$ we mean its set of edges. A graph is cubic if each of its vertices is connected to exactly three other ones. Our NP-hardness proofs rely on reductions from variants of the Independent Set and X3C problems, both known to be NP-complete [18, 19].

Definition 1. *In the* Cubic Independent Set *problem we are given a cubic graph G and an integer h; we ask if G has an independent set of size h (i.e., a set of h vertices such that no two of them are connected).*

Definition 2. *In the* Restricted Exact Cover by 3-Sets *problem (RX3C) we are given a universe X of $3n$ elements and a family $\mathcal{S}$ of $3n$ size-3 subsets of X. Each element from X appears in exactly three sets from $\mathcal{S}$. We ask if it is possible to choose n sets from $\mathcal{S}$ whose union is X.*

3 ADDING APPROVALS

For the case of adding approvals, all our bribery problems (priced and unpriced, both for the CI and VI domains) remain solvable in polynomial time. Yet, compared to the unrestricted setting, our algorithms require more care. For example, in the unrestricted case it suffices to simply add approvals for the preferred candidate [16] (choosing the voters where they are added in the order of increasing costs for the priced variant); a similar approach works for the VI case, but with a different ordering of the voters.

Theorem 1. *AV-\$AddApprovals-VI-Bribery* $\in$ P.

The CI case introduces a different complication. Now, adding an approval for the preferred candidate in a given vote also requires adding approvals for all those between him or her and the original approval set. Thus, in addition to bounding the bribery's cost, we also need to track the candidates whose scores increase.

Theorem 2. *AV-\$AddApprovals-CI-Bribery* $\in$ P.

Proof. Our input consists of an election $E = (C, V)$, committee size k, preferred candidate $p \in C$, budget B, and the information about the costs of all the possible operations (i.e., for each voter and each candidate that he or she does not approve, we have the price for adding this candidate to the voter's ballot). Without loss of generality, we assume that $C = \{\ell_{m'}, \dots, \ell_1, p, r_1, \dots, r_{m''}\}$, $V = \{v_1, \dots, v_n\}$, each voter approves at least one candidate,[2] and the election is CI with respect to the order:

$$\rhd = \ell_{m'} \cdots \ell_2 \; \ell_1 \; p \; r_1 \; r_2 \cdots r_{m''}.$$

We start with a few observations. First, if a voter already approves p then there is no point in adding any approvals to his or her ballot. Second, if some voter does not approve p, then we should either not add any approvals to his or her ballot, or add exactly those approvals that are necessary to ensure that p gets one. For example, if some voter has approval ballot $\{r_3, r_4, r_5\}$ then we may either choose to leave it intact or to extend it to $\{p, r_1, r_2, r_3, r_4, r_5\}$. We let $L = \{\ell_{m'}, \dots, \ell_1\}$ and $R = \{r_1, \dots, r_{m''}\}$, and we partition the voters into three groups, V_ℓ, V_p, and V_r, as follows:

(1) V_p contains all the voters who approve p,
(2) V_ℓ contains the voters who approve members of L only,
(3) V_r contains the voters who approve members of R only.

Our algorithm proceeds as follows (by guessing we mean iteratively trying all possibilities; Steps 3 and 4 will be described later):

(1) Guess the numbers x_ℓ and x_r of voters from V_ℓ and V_r whose approval ballots will be extended to approve p.
(2) Guess the numbers t_ℓ and t_r of candidates from L and R that will end up with higher approval scores than p (we must have $t_\ell + t_r < k$ for p to join a winning committee).
(3) Compute the lowest cost of extending exactly x_ℓ votes from V_ℓ to approve p, such that at most t_ℓ candidates from L end up with more than $\mathrm{score}_E(p) + x_\ell + x_r$ points (i.e., with score higher than p); denote this cost as B_ℓ.
(4) Repeat the above step for the x_r voters from V_r, with at most t_r candidates obtaining more than $\mathrm{score}_E(p) + x_\ell + x_r$ points; denote the cost of this operation as B_r.
(5) If $B_\ell + B_r \le B$ then accept (reject if no choice of x_ℓ, x_r, t_ℓ, and t_r leads to acceptance).

One can verify that this algorithm is correct (assuming we know how to perform Steps 3 and 4).

Next we describe how to perform Step 3 in polynomial time (Step 4 is handled analogously). We will need some additional notation. For each $i \in [m']$, let $V_\ell(i)$ consist exactly of those voters from V_ℓ whose approval ballots include candidate ℓ_i but do not include ℓ_{i-1} (in other words, voters in $V_\ell(i)$ have approval ballots of the form $[\ell_j, \ell_i]$, where $j \ge i$). Further, for each $i \in [m']$ and each $e \in [|V_\ell(i)|]_0$ let $\mathrm{cost}(i, e)$ be the lowest cost of extending e votes from $V_\ell(i)$ to approve p (and, as a consequence, to also approve candidates $\ell_{i-1}, \dots, \ell_1$). If $V_\ell(i)$ contains fewer than e voters then $\mathrm{cost}(i, e) = +\infty$. For each $e \in [x_\ell]_0$, we define $S(e) = \mathrm{score}_E(p) + e + x_r$. Finally, for each $i \in [m']$, $e \in [x_\ell]_0$, and $t \in [t_\ell]_0$ we define function $f(i, e, t)$ so that:

> $f(i, e, t)$ = the lowest cost of extending exactly e votes from $V_\ell(1) \cup \cdots \cup V_\ell(i)$ (to approve p) such that at most t candidates among $\ell_1, \dots, \ell_i$ end up with more than $S(e)$ points (function f takes value $+\infty$ if doing so is impossible).

Our goal in Step 3 of the main algorithm is to compute $f(m', x_\ell, t_\ell)$, which we do via dynamic programming. To this end, we observe that the following recursive equation holds (let $\chi(i, e)$ be 1 if $\mathrm{score}_E(\ell_i) > S(e)$ and let $\chi(i, e)$ be 0 otherwise; we explain the idea of the equation below):

$$f(i, e, t) = \min_{e' \in [e]_0} \big(\mathrm{cost}(i, e') + f(i-1, e-e', t-\chi(i, e))\big).$$

The intuition behind this equation is as follows. We consider each possible number $e' \in [e]_0$ of votes from $V_\ell(i)$ that can be extended to approve p. The lowest cost of extending the votes of e' voters from $V_\ell(i)$ is, by definition, $\mathrm{cost}(i, e')$. Next, we still need to extend $e - e'$ votes from $V_\ell(i-1), \dots, V_\ell(1)$ and, while doing so, we need to ensure that at most t candidates end up with at most $S(e)$ points. Candidate ℓ_i cannot get any additional approvals from voters $V_\ell(i-1), \dots, V_\ell(1)$, so he or she exceeds this value exactly if $\mathrm{score}_E(\ell_i) > S(e)$ or, equivalently, if $\chi(i, e) = 1$. This means that we have to ensure that at most $t - \chi(i, e)$ candidates among $\ell_{i-1}, \dots, \ell_1$ end up with at most $S(e)$ points. However, since we extend e' votes from $V_\ell(i)$, we know that candidates $\ell_{i-1}, \dots, \ell_1$ certainly obtain e' additional points (as compared to the input election). Thus we need to ensure that at most $t - \chi(i, e)$ of them end up with score at most $S(e-e')$ after extending the votes from $V_\ell(1) \cup \ldots \cup V_\ell(i-1)$. This is ensured by the $f(i-1, e-e', t-\chi(i, e))$ component in the equation (which also provides the lowest cost of the respective operations).

[2] Without this assumption we could still make our algorithm work. We would guess the number of voters who do not approve any candidates to approve p alone (we would choose these voters to minimize the cost of adding these approvals). Then we would continue as in the proof, but knowing that none of the voters in the group can be bribed further.

Using the above formula, the fact that $f(1, e, t)$ can be computed easily for all values of e and t, and standard dynamic programming techniques, we can compute $f(m', x_\ell, t_\ell)$ in polynomial time. This suffices for completing Step 3 of the main algorithm and we handle Step 4 analogously. Since all the steps of can be performed in polynomial time, the proof is complete. □

Both above theorems also apply to the cases where we can only add approvals for the preferred candidate. The algorithm from Theorem 1 is designed to do just that, and for the algorithm from Theorem 2 we can set the price of adding other approvals to be $+\infty$.

4 DELETING APPROVALS

The case of deleting approvals is more intriguing. Roughly speaking, in the unrestricted setting it suffices to delete approvals from sufficiently many candidates that have higher scores than p, for whom doing so is least expensive [16]. The same general strategy works for the VI case because we still can delete approvals for different candidates independently.

Theorem 3. *AV-\$DelApprovals-VI-Bribery* $\in$ P.

Proof. Let our input consist of an election $E = (C, V)$, preferred candidate $p \in C$, committee size k, and budget B. We assume that $V = \{v_1, \ldots, v_n\}$ and the election is VI with respect to ordering the voters by their indices. Let $s = \mathrm{score}_E(p)$ be the score of p prior to any bribery. We refer to the candidates with score greater than s as *superior*. Since it is impossible to increase the score of p by deleting approvals, we need to ensure that the number of superior candidates drops to at most $k - 1$.

For each superior candidate c, we compute the lowest cost for reducing his or her score to exactly s. Specifically, for each such candidate c we act as follows. Let $t = \mathrm{score}_E(c) - s$ be the number of c's approvals that we need to delete and let $v_a, v_{a+1}, \ldots, v_b$ be the interval of voters that approve c. For each $i \in [t]_0$ we compute the cost of deleting c's approvals among the first i and the last $t - i$ voters in the interval (these are the only operations that achieve our goal and maintain the VI property of the election); we store the lowest of these costs as "the cost of c."

Let S be the number of superior candidates (prior to any bribery). We choose $S - (k - 1)$ of them with the lowest costs. If the sum of these costs is at most B then we accept and, otherwise, we reject. □

For the CI case, our problem turns out to be NP-complete. Intuitively, the reason for this is that in the CI domain deleting an approval for a given candidate requires either deleting all the approvals to the left or all the approvals to the right on the societal axis. Indeed, our main trick is to introduce approvals that must be deleted (at zero cost), but doing so requires choosing whether to delete their left or their right neighbors (at nonzero cost). This result is our first example of a complexity reversal.

Theorem 4. *AV-\$DelApprovals-CI-Bribery is* NP*-complete.*

Proof. We give a reduction from RX3C. Let $I = (X, \mathcal{S})$ be the input instance, where $X = \{x_1, \ldots, x_{3n}\}$ is the universe and $\mathcal{S} = \{S_1, \ldots, S_{3n}\}$ is a family of size-3 subsets of X. By definition, each element of X belongs to exactly three sets from $\mathcal{S}$. We form an instance of AV-\$DelApprovals-CI-Bribery as follows.

We have the preferred candidate p, for each universe element $x_i \in X$ we have corresponding universe candidate x_i, for each set $S_j \in \mathcal{S}$ we have set candidate s_j, and we have set D of $2n$ dummy candidates (where each individual one is denoted by $\diamond$). Let C be the set of just-described $8n + 1$ candidates and let $S = \{s_1, \ldots, s_{3n}\}$ contain the set candidates. We fix the societal axis to be:

$$\rhd = \underbrace{\overbrace{s_1 \cdots s_{3n}}^{3n} \overbrace{\diamond \cdots \diamond}^{2n} \overbrace{x_1 \cdots x_{3n}}^{3n} p}_{8n+1}$$

Next, we form the voter collection V:

(1) For each candidate in $S \cup D \cup \{p\}$, we have two voters that approve exactly this candidate. We refer to them as the *fixed voters* and we set the price for deleting their approvals to be $+\infty$. We refer to their approvals as *fixed*.
(2) For each set $S_j = \{x_a, x_b, x_c\}$, we form three *solution voters*, $v(s_j, x_a)$, $v(s_j, x_b)$, and $v(s_j, x_c)$, with approval sets $[s_j, x_a]$, $[s_j, x_b]$, and $[s_j, x_c]$, respectively. For a solution voter $v(s_i, x_d)$, we refer to the approvals that s_i and x_d receive as *exterior*, and to all the other ones as *interior*. The cost for deleting each exterior approval is one, whereas the cost for deleting the interior approvals is zero. Altogether, there are $9n$ solution voters.

To finish the construction, we set the committee size $k = n + 1$ and the budget $B = 9n$. Below, we list the approval scores prior to any bribery (later we will see that in successful briberies one always deletes all the interior approvals):

(1) p has 2 fixed approvals,
(2) each universe candidate has 3 exterior approvals (plus some number of interior ones),
(3) each set candidate has 3 exterior approvals and 2 fixed ones (plus some number of interior ones), and
(4) each dummy candidate has 2 fixed approvals (and $9n$ interior ones).

We claim that there is a bribery of cost at most B that ensures that p belongs to some winning committee if and only if I is a yes-instance of RX3C. For the first direction, let us assume that I is a yes-instance and let $\mathcal{T}$ be a size-n subset of $\mathcal{S}$ such that $\bigcup_{S_i \in \mathcal{T}} S_i = X$ (i.e., $\mathcal{T}$ is the desired exact cover). We perform the following bribery: First, for each solution voter we delete all his or her interior approvals. Next, to maintain the CI property (and to lower the scores of some candidates), for each solution voter we delete one exterior approval. Specifically, for each set $S_j = \{x_a, x_b, x_c\}$, if S_j belongs to the cover (i.e., if $S_i \in \mathcal{T}$) then we delete the approvals for x_a, x_b, and x_c in $v(s_j, x_a)$, $v(s_j, x_b)$, and $v(s_j, x_c)$, respectively; otherwise, i.e., if $S_j \notin \mathcal{T}$, we delete the approvals for s_j in these votes. As a consequence, all the universe candidates end up with two exterior approvals each, the n set candidates corresponding to the cover end up with three approvals each (two fixed ones and one exterior), the $2n$ remaining set candidates and all the dummy candidates end up with two fixed approvals each. Since p has two approvals, the committee size is $n + 1$, and only n candidates have score higher than p, p belongs to some winning committee (and the cost of the bribery is B).

For the other direction, let us assume that there is a bribery with cost at most B that ensures that p belongs to some winning committee. It must be the case that this bribery deletes exactly one exterior approval from each solution voter. Otherwise, since there are $9n$ solution voters and the budget is also $9n$, some solution voter would keep both his or her exterior approvals, as well as all the interior ones. This means that after the bribery there would be at least $2n$ dummy candidates with at least three points each. Then, p would not belong to any winning committee. Thus, each solution voter deletes exactly one exterior approval, and we may assume that he or she also deletes all the interior ones (this comes at zero cost and does not decrease the score of p).

By the above discussion, we know that all the dummy candidates end up with two fixed approvals, i.e., with the same score as p. Thus, for p to belong to some winning committee, at least $5n$ candidates among the set and universe ones also must end up with at most two approvals (at most n candidates can have score higher than p). Let x be the number of set candidates whose approval score drops to at most two, and let y be the number of such universe candidates. We have that:

$$0 \le x \le 3n, \qquad 0 \le y \le 3n, \qquad \text{and} \qquad x + y \ge 5n. \quad (1)$$

Prior to the bribery, each set candidate has five non-interior approvals (including three exterior approvals) so bringing his or her score to at most two costs three units of budget. Doing the same for a universe candidate costs only one unit of budget, as universe candidates originally have only three non-interior approvals. Since our total budget is $9n$, we have:

$$3x + y \le 9n. \quad (2)$$

Together, inequalities (1) and (2) imply that $x = 2n$ and $y = 3n$. That is, for each universe candidate x_i there is a solution voter $v(s_j, x_d)$ who is bribed to delete the approval for x_d (and, as a consequence of our previous discussion, who is not bribed to delete the approval for s_j). We call such solution voters *active* and we define a family:

$$\mathcal{T} = \{S_j \mid s_j \text{ is approved by some active solution voter}\}.$$

We claim that $\mathcal{T}$ is an exact cover for the RX3C instance I. Indeed, by definition of active solution voters we have that $\bigcup_{S_i \in \mathcal{T}} S_i = X$. Further, it must be the case that $|\mathcal{T}| = n$. This follows from the observation that if some solution voter is active then his or her corresponding set candidate s_j has at least three approvals after the bribery (each set candidate receives exterior approvals from exactly three solution voters and these approvals must be deleted if the candidate is to end up with score two; this is possible only if all the three solution voters are not active). Since exactly $2n$ set candidates must have their scores reduced to two, it must be that $3n - |\mathcal{T}| = 2n$, so $|\mathcal{T}| = n$. This completes the proof. □

The above proof strongly relies on using $0/1/+\infty$ prices. The case of unit prices remains open and we believe that resolving it might be quite challenging.

5 SWAPPING APPROVALS

In some sense, bribery by swapping approvals is our most interesting scenario because there are cases where a given problem has the same complexity both in the unrestricted setting and for some structured domain (and this happens both for tractability and NP-completeness), as well as cases where the unrestricted variant is tractable but the structured one is not or the other way round.

5.1 Approval Swaps to the Preferred Candidate

Let us first consider a variant of AV-SwapApprovals-Bribery where each unit operation moves an approval from some candidate to the preferred one. We call operations of this form SwapApprovals to p. In the unrestricted setting, this problem is in P for unit prices but is NP-complete if the prices are arbitrary. For the CI and VI domains, the problem can be solved in polynomial time for both types of prices. While for the CI domain this is not so surprising—indeed, in this case possible unit operations are very limited—the VI case requires quite some care.

Theorem 5. *AV-\$SwapApprovals to p-CI-Bribery* $\in$ P.

Our algorithm for the VI case is based on dynamic programming (expressed as searching for a shortest path in a certain graph) and relies on the fact that due to the VI property we avoid performing the same unit operations twice.

Theorem 6. *AV-\$SwapApprovals to p-VI-Bribery* $\in$ P.

5.2 Arbitrary Swaps

Next, we consider the full variant of bribery by swapping approvals. For the unrestricted domain, the problem is NP-complete for general prices, but admits a polynomial-time algorithm for unit ones [16]. For the CI domain, NP-completeness holds even for the latter.

Remark 1. *The model of unit prices, applied directly to the case of SwapApprovals-CI-Bribery, is somewhat unintuitive. For example, consider societal axis* $c_1 \rhd c_2 \rhd \cdots \rhd c_{10}$ *and an approval set* $[c_3, c_5]$*. The costs of swap operations that transform it into, respectively,* $[c_4, c_6]$*,* $[c_5, c_7]$*, and* $[c_6, c_8]$ *are* 1, 2, *and* 3*, as one would naturally expect. Yet, the cost of transforming it into, e.g.,* $[c_8, c_{10}]$ *would also be* 3 *(move an approval from* c_3 *to* c_8*, from* c_4 *to* c_9*, and from* c_5 *to* c_{10}*), which is not intuitive. Instead, it would be natural to define this cost to be* 5 *(move the interval by* 5 *positions to the right). Our proof of Theorem 7 works without change for both these interpretations of unit prices.*

Theorem 7. *AV-SwapApprovals-CI-Bribery is* NP*-complete.*

Proof. We give a reduction from Cubic Independent Set. Let G be our input graph, where $V(G) = \{c_1, \ldots, c_n\}$ and $E(G) = \{e_1, \ldots, e_L\}$, and let h be the size of the desired independent set. We construct the corresponding AV-SwapApprovals-CI-Bribery instance as follows.

Let $B = 3h$ be our budget and let $t = B + 1$ be a certain parameter (which we interpret as "more than the budget"). We form the candidate set $C = V(G) \cup \{p\} \cup F \cup D$, where p is the preferred candidate, F is a set of $t(n+1)$ filler candidates, and D is a set of t dummy candidates. Altogether, there are $t(n+2)+n+1$ candidates. We denote individual filler candidates by $\diamond$ and individual dummy candidates by $\bullet$; we fix the societal axis to be:

$$\rhd = \underbrace{\overbrace{\diamond\cdots\diamond}^{t} c_1 \overbrace{\diamond\cdots\diamond}^{t} c_2 \diamond\cdots\diamond c_{n-1} \overbrace{\diamond\cdots\diamond}^{t} c_n \overbrace{\diamond\cdots\diamond}^{t} \overbrace{\bullet\cdots\bullet}^{t} p}_{t(n+2)+n+1}$$

For each positive integer i and each candidate c, we write $\mathrm{prec}_i(c)$ to mean the i-th candidate preceding c in $\rhd$. Similarly, we write $\mathrm{succ}_i(c)$ to denote the i-th candidate after c. We introduce the following voters:

(1) For each edge $e_i = \{c_a, c_b\}$ we add an edge voter $v_{a,b}$ with approval set $[c_a, c_b]$. For each vertex $c_i \in V(G)$, we write $V(c_i)$ to denote the set of the three edge voters corresponding to the edges incident to c_i.
(2) Recall that $L = |E(G)|$. For each vertex candidate $c_i \in V(G)$, we add sufficiently many voters with approval set $[\mathrm{prec}_t(c_i), \mathrm{succ}_t(c_i)]$, so that, together with the score from the edge voters, c_i ends up with L approvals.
(3) We add $L - 3$ voters that approve p.
(4) For each group of t consecutive filler candidates, we add $L + 4t$ filler voters, each approving all the candidates in the group.

Altogether, p has score $L - 3$, all vertex candidates have score L, the filler candidates have at least $L + 4t$ approvals each, and the dummy candidates have score 0. We set the committee size to be $k = t(n + 1) + (n - h) + 1$. Prior to any bribery, each winning committee consists of $t(n+1)$ filler candidates and $(n-h)+1$ vertex ones (chosen arbitrarily). This completes our construction.

Let us assume that G has a size-h independent set and denote it with S. For each $c_i \in S$ and each edge $e_t = \{c_i, c_j\}$, we bribe edge voter $v_{i,j}$ to move an approval from c_i to a filler candidate right next to c_j. This is possible for each of the three edges incident to c_i because S is an independent set. As a consequence, each vertex from S ends up with $L - 3$ approvals. Thus only $n - h$ vertex candidates have score higher than p and, so, there is a winning committee that includes p.

For the other direction, let us assume that it is possible to ensure that p belongs to some winning committee via a bribery of cost at most B. Let us consider the election after some such bribery was executed. First, we note that all the filler candidates still have scores higher than $L+3t$ (this is so because decreasing a candidate's score always has at least unit cost and $B < t$). Similarly, p still has score $L - 3$ because increasing his or her score, even by one, costs at least t (indeed, p is separated from the other candidates by t dummy candidates). Since p belongs to some winning committee, this means that at least h vertex voters must have ended up with score at most $L - 3$. In fact, since our budget is $B = 3h$, a simple counting argument shows that exactly h of them have score exactly $L - 3$, and all the other ones still have score L. Let S be the set of vertex candidates with score $L - 3$. The only way to decrease the score of a vertex candidate c_i from L to $L - 3$ by spending three units of the budget is to bribe each of the three edge voters from $V(c_i)$ to move an approval from c_i to a filler candidate. However, if we bribe some edge voter $v_{i,j}$ to move an approval from c_i to a filler candidate, then we cannot bribe that same voter to also move an approval away from c_j (this would either cost more than t units of budget or would break the CI condition). Thus it must be the case that the candidates in S correspond to a size-h independent set for G. □

For the VI domain, the complexity of our problem for unit prices remains open, but for arbitrary prices we show that it is NP-complete. Our proof works even for the single-winner setting. In the unrestricted domain, the single-winner variant is in P [11].

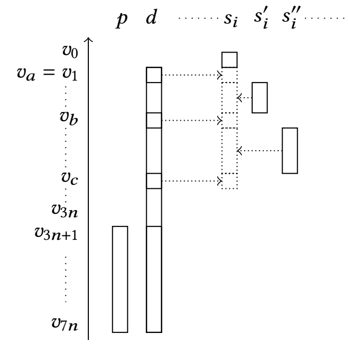

Figure 1: Illustration of the election from the proof of Theorem 8, for the case where $S_i = \{x_a, x_b, x_c\}$, where $a = 1 < b < c$. Each row corresponds to a voter and each column corresponds to a candidate. Solid boxes show approvals prior bribery and dotted ones show approval moves.

Theorem 8. *AV-\$SwapApprovals-VI-Bribery is* NP*-complete, even for the single-winner case (i.e., for committees of size one).*

Proof. We give a reduction from RX3C. Let $I = (X, \mathcal{S})$ be an instance of RX3C, where $X = \{x_1, \dots, x_{3n}\}$ is a universe and $\mathcal{S} = \{S_1, \dots, S_{3n}\}$ is a family of size-3 subsets of X (recall that each element from X belongs to exactly three sets from $\mathcal{S}$). We form a single-winner approval election with $7n + 1$ voters $V = \{v_0, v_1, \dots, v_{7n}\}$ and the following candidates:

(1) We have the preferred candidate p and the (to be defeated) current winner d.
(2) For each set $S_i \in \mathcal{S}$ we have candidates s_i, s_i', and s_i''.

The approvals for these candidates, and the costs of moving them, are as follows (if we do not explicitly list the cost of moving some approval from a given candidate to another, then it is $+\infty$, i.e., this swap is impossible; the construction is illustrated in Figure 1):

(1) Candidate p is approved by $4n$ voters, $v_{3n+1}, \dots, v_{7n}$.
(2) Candidate d is approved by $7n$ voters, $v_1, \dots, v_{7n}$. For each set $S_i = \{x_a, x_b, x_c\}$, where $a < b < c$, the cost of moving v_a's approval from d to s_i is 1, and the costs of moving v_b's and v_c's approvals from d to s_i is 0.
(3) For each set $S_i = \{x_a, x_b, x_c\}$, where $a < b < c$, we have the following approvals. Candidate s_i is approved by voter v_{a-1}, candidate s_i' is approved by voters $v_{a+1}, \dots, v_{b-1}$, and candidate s_i'' is approved by voters $v_{b+1}, \dots, v_{c-1}$. The cost of moving the approvals from s_i' or from s_i'' to s_i is 0.

One can verify that this election has the VI property for the natural order of the voters (i.e., for $v_0 \rhd \cdots \rhd v_{7n}$). Candidate d has $7n$ approvals, p has $4n$ approvals, and every other candidate has at most $3n + 1$ approvals. We claim that it is possible to ensure that p becomes a winner of this election by approval-moves of cost at most $B = n$ (such that the election still has the VI property after these moves) if and only if I is a yes-instance of RX3C.

For the first direction, let us assume that I is a yes-instance and that $R \subseteq [3n]$ is a size-n set such that $\bigcup_{i \in R} S_i = X$ (naturally, for each $t, \ell \in R$, sets S_t and S_ℓ are disjoint). It is possible to ensure that p becomes a winner by making, for each $S_i = \{x_a, x_b, x_c\}$ such that $i \in R$ and $a < b < c$, the following swaps:

(1) For each $j \in \{a, b, c\}$, we move v_j's approval from d to s_i (the cost of moving v_a's approval is 1, the two other moves have cost 0).
(2) For each $j \in \{a+1, \dots, b-1\}$, we move v_j's approval from s'_i to s_i (at cost 0).
(3) For each $j \in \{b+1, \dots, c-1\}$, we move v_j's approval from s''_i to s_i (at cost 0).

In total, these moves cost n and, since R corresponds to a cover of X, we have that: (a) p is approved by $4n$ voters, (b) d is approved by $4n$ voters, and (c) every other candidate is approved by at most $3n + 1$ voters. Consequently, p is among tied winners of this election.

For the other direction, let us assume that there is a sequence of approval moves that costs at most n and ensures that p is a winner. Since all the moves of approvals from and to p have cost $+\infty$, this means that every candidate ends up with at most $4n$ points. Thus d loses at least $3n$ approvals. No matter what swaps we do, for each $i \in [3n]$ each of s_i, s'_i and s''_i ends up with at most $3n + 1$ approvals so we do not need to count their scores carefully (but we do need to take the VI condition into account for these candidates).

Candidate d can lose approvals only due to voters $v_1, \dots, v_{3n}$ moving them to candidates in $\{s_1, \dots, s_{3n}\}$. Let us consider some candidate s_i such that some voter v_j moves an approval from d to s_i and let $a < b < c$ be such that $S_i = \{x_a, x_b, x_c\}$. Due to the costs of moving approvals, it must be that $j \in \{a, b, c\}$. In fact, we claim that all three voters v_a, v_b, v_c move approvals to s_i, voters $v_{a+1}, \dots, v_{b-1}$ move approvals from s'_i to s_i, and voters $v_{b+1}, \dots, v_{c-1}$ move approvals from s''_i to s_i. This is so, because if those voters $v_a, \dots, v_j$ would not move their approvals, then—due to the fact that s_i is approved by voter v_{a-1} (and this approval cannot move given our budget)—the approvals for s_i would not satisfy the VI property. Further, voters $v_{j+1}, \dots, v_c$ also need to move their approvals due to a counting argument: The cost of moving v_a's approval from d to s_i is 1. If we did not move v_c's approvals from d to s_i, then it would mean that (globally in our bribery) the average cost of moving an approval from d to some candidate in $\{s_1, \dots, s_{3n}\}$ would be higher than 1/3. But since our budget is n and we need to move $3n$ approvals from d to these candidates, this is impossible.

Let $R = \{i \in [3n] \mid \text{some voter moves an approval from candidate } d \text{ to } s_i\}$. By the preceding paragraph, R contains n elements and for each two $i, j \in R$ it must be that sets S_i and S_j are disjoint. Hence, I is a yes-instance. □

6 DESTRUCTIVE BRIBERY

We conclude by considering destructive variants of our problems, where the goal is to ensure that a given candidate, often denoted d, does not belong to any winning committee. We use the same bribery actions, except that now we also consider a variant of swapping approvals where we can only move approvals away from d.

The destructive variant has been studied for the unrestricted setting by Yang [32], for the unpriced cases of adding and deleting approvals. Thus we first establish its complexity also for the priced cases and for swapping approvals. The complexity stays the same as for the constructive variants (the theorem below includes the results of Yang [32] as special cases).

THEOREM 9. *For unit prices, all destructive variants of our bribery problems for the unrestricted setting are in* P. *For arbitrary prices, the cases of adding and deleting approvals are in* P, *but destructive variants AV-$SwapApprovalsAwayFromD-Bribery and AV-$SwapApprovals-Bribery are* NP*-complete.*

For the VI case, we also obtain (or, fail to obtain) almost the same results as in the constructive case (for AV-$SwapApprovals-VI-Bribery we use the same proof as in the constructive case, except d is the distinguished candidate and p has one extra approval).

THEOREM 10. *Destructive variants of AV-$AddApprovals-VI-Bribery and AV-$DelApprovals-VI-Bribery are in* P. *Destructive variant of AV-$SwapApprovals-VI-Bribery is* NP*-complete.*

The case of CI preferences appears to be the most challenging one. Not only do we obtain fewer results than in the constructive setting, but those that we do obtain are less satisfying. Let us illustrate this with AV-$AddApprovals-CI-Bribery. We show that the problem is NP-complete, but to do so, we use a somewhat unappealing trick. Namely, we include some voters who initially do not approve any candidates and we set their price functions so that we can choose one out of four candidates, possibly located far apart in the societal axis, to whom these voters add an approval. Since we did not put extra conditions on the price functions, this is formally correct, but is intuitively unappealing. We also need similar tricks in two other NP-completenss proofs in this section. For example, for the case of swapping approvals away from d we use voters that approve only a single candidate, so we can move this approval arbitrarily (up to constraints implemented with the price function). Interestingly, if we required each voter to approve at least two candidates, the problem would be in P.

THEOREM 11. *Destructive variants of AV-$AddApprovals-CI-Bribery, AV-$SwapApprovalsAwayFromD-CI-Bribery, and AV-$SwapApprovals-CI-Bribery are* NP*-complete. Destructive variant of AV-SwapApprovalsAwayFromD-CI-Bribery is in* P.

7 SUMMARY

We have studied bribery in multiwinner approval elections, for the case of candidate interval (CI) and voter interval (VI) preferences. Depending on the setting, our problem can either be easier, harder, or equally difficult as in the unrestricted domain. It would be interesting to extend our work by considering different voting rules (in particular, the Approval-Based Chamberlin–Courant rule [2, 8, 25]) and by seeking parameterized complexity results.

ACKNOWLEDGMENTS

We want to thank the reviewers for their helpful comments. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No 101002854). DK acknowledges the support of the GAČR project No. 22-19557S.

erc
European Research Council
Established by the European Commission

REFERENCES

[1] D. Baumeister and T. Hogrebe. 2021. On the Complexity of Predicting Election Outcomes and Estimating Their Robustness. In *Proceedings of EUMAS-21*. 228–244.

[2] N. Betzler, A. Slinko, and J. Uhlmann. 2013. On the Computation of Fully Proportional Representation. *Journal of Artificial Intelligence Research* 47 (2013), 475–519.

[3] D. Black. 1958. *The Theory of Committees and Elections*. Cambridge University Press.

[4] N. Boehmer, R. Bredereck, P. Faliszewski, and R. Niedermeier. 2021. Winner Robustness via Swap- and Shift-Bribery: Parameterized Counting Complexity and Experiments. In *Proceedings of IJCAI-21*. 52–58.

[5] K. Booth and G. Lueker. 1976. Testing for the Consecutive Ones Property, Interval Graphs, and Graph Planarity Using PQ-Tree Algorithms. *J. Comput. System Sci.* 13, 3 (1976), 335–379.

[6] F. Brandt, M. Brill, E. Hemaspaandra, and L. Hemaspaandra. 2015. Bypassing Combinatorial Protections: Polynomial-Time Algorithms for Single-Peaked Electorates. *Journal of Artificial Intelligence Research* 53 (2015), 439–496.

[7] R. Bredereck, P. Faliszewski, A. Kaczmarczyk, R. Niedermeier, P. Skowron, and N. Talmon. 2021. Robustness among multiwinner voting rules. *Artificial Intelligence* 290 (2021), 103403.

[8] B. Chamberlin and P. Courant. 1983. Representative Deliberations and Representative Decisions: Proportional Representation and the Borda Rule. *American Political Science Review* 77, 3 (1983), 718–733.

[9] E. Elkind, P. Faliszewski, S. Gupta, and S. Roy. 2020. Algorithms for Swap and Shift Bribery in Structured Elections. In *Proceedings of AAMAS-20*. 366–374.

[10] E. Elkind and M. Lackner. 2015. Structure in Dichotomous Preferences. In *Proceedings of IJCAI-15*. 2019–2025.

[11] P. Faliszewski. 2008. Nonuniform Bribery (Short Paper). In *Proceedings of AAMAS-08*. 1569–1572.

[12] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. 2009. How Hard is Bribery in Elections? *Journal of Artificial Intelligence Research* 35 (2009), 485–532.

[13] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. 2011. The Shield that Never Was: Societies with Single-Peaked Preferences are More Open to Manipulation and Control. *Information and Computation* 209, 2 (2011), 89–107.

[14] P. Faliszewski and J. Rothe. 2015. Control and Bribery in Voting. In *Handbook of Computational Social Choice*, F. Brandt, V. Conitzer, U. Endriss, J. Lang, and A. D. Procaccia (Eds.). Cambridge University Press, Chapter 7.

[15] P. Faliszewski, P. Skowron, A. Slinko, and N. Talmon. 2017. Multiwinner Voting: A New Challenge for Social Choice Theory. In *Trends in Computational Social Choice*, U. Endriss (Ed.). AI Access Foundation.

[16] P. Faliszewski, P. Skowron, and N. Talmon. 2017. Bribery as a Measure of Candidate Success: Complexity Results for Approval-Based Multiwinner Rules. In *Proceedings of AAMAS-17*. 6–14.

[17] Z. Fitzsimmons and E. Hemaspaandra. 2015. Complexity of Manipulative Actions When Voting with Ties. In *Proceedings of ADT-15*. 103–119.

[18] M. Garey and D. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company.

[19] T. Gonzalez. 1985. Clustering to Minimize the Maximum Intercluster Distance. *Theoretical Computer Science* 38 (1985), 293–306.

[20] M. Lackner and P. Skowron. 2021. *Approval-Based Committee Voting: Axioms, Algorithms, and Applications*. Technical Report arXiv:2007.01795 [cs.GT]. arXiv.org.

[21] K. Magiera and P. Faliszewski. 2017. How Hard is Control in Single-Crossing Elections? *Autonomous Agents and Multiagent Systems* 31, 3 (2017), 606–627.

[22] V. Menon and K. Larson. 2016. Reinstating Combinatorial Protections for Manipulation and Bribery in Single-Peaked and Nearly Single-Peaked Electorates. In *Proceedings of AAAI-16*. 565–571.

[23] J. Mirrlees. 1971. An Exploration in the Theory of Optimal Income Taxation. *Review of Economic Studies* 38 (1971), 175–208.

[24] D. Peters and M. Lackner. 2020. Preferences Single-Peaked on a Circle. *Journal of Artificial Intelligence Research* 68 (2020), 463–502.

[25] A. Procaccia, J. Rosenschein, and A. Zohar. 2008. On the Complexity of Achieving Proportional Representation. *Social Choice and Welfare* 30, 3 (2008), 353–362.

[26] K. Roberts. 1977. Voting Over Income Tax Schedules. *Journal of Public Economics* 8, 3 (1977), 329–340.

[27] I. Schlotter, P. Faliszewski, and E. Elkind. 2017. Campaign Management Under Approval-Driven Voting Rules. *Algorithmica* 77, 1 (2017), 84–115.

[28] D. Shiryaev, L. Yu, and E. Elkind. 2013. On Elections with Robust Winners. In *Proceedings of AAMAS-13*. 415–422.

[29] T. Walsh. 2007. Uncertainty in Preference Elicitation and Aggregation. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*. 3–8.

[30] L. Xia. 2012. Computing the Margin of Victory for Various Voting Rules. In *Proceedings of EC-12*. 982–999.

[31] Y. Yang. 2017. On the Complexity of Borda Control in Single-Peaked Elections. In *Proceedings of AAMAS-17*. 1178–1186.

[32] Y. Yang. 2020. On the Complexity of Destructive Bribery in Approval-Based Multi-winner Voting. In *Proceedings of AAMAS-20*. 1584–1592.