On Manipulation in Multi-Winner Elections Based on Scoring Rules

Svetlana Obraztsova
Steklov Institute of Mathematics
St. Petersburg, Russia
and National Technical
University of Athens, Greece
svetlana.obraztsova@gmail.com

Yair Zick
School of Physical and
Mathematical Sciences
Nanyang Technological
University, Singapore
yair0001@ntu.edu.sg

Edith Elkind
School of Physical and
Mathematical Sciences
Nanyang Technological
University, Singapore
eelkind@ntu.edu.sg

ABSTRACT

Multi-winner elections model scenarios where voters must select a fixed-size group of candidates based on their individual preferences. In such scenarios, it is often the case that voters are incentivized to *manipulate*, i.e. misreport their preferences in order to obtain a better outcome. In this paper, we study the complexity of manipulating multi-winner elections under scoring rules, with a particular focus on the role of tie-breaking rules. We consider both lexicographic tie-breaking rules, which break ties according to a fixed ordering of the candidates, and a natural randomized tie-breaking rule. We describe polynomial-time manipulation algorithms for several special cases of our problem. Specifically, we show that finding a successful manipulation is easy if the underlying voting rule is k-Approval or the number of candidates to be elected is bounded by a constant (for both types of tie-breaking rules), as well as if the manipulator's utility function only takes values in $\{0, 1\}$ and the ties are broken in the manipulator's favor.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems; F.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity

Keywords

voting, multi-winner elections, tie-breaking rules

1. INTRODUCTION

Preference aggregation is an important component of collective decision-making, and has been actively studied by the multi-agent research community in recent years. It is usually assumed that each agent's preferences are represented by a ranking of all available alternatives, and the agents aim to select a single alternative. However, there are also settings where the goal is to choose a fixed-size *set* of alternatives. Perhaps the most natural example of such a scenario is provided by parliamentary elections, where the voters need to choose a fixed-size assembly that will represent them in the best possible way. There are a number of voting rules that have been designed for this purpose, such as Single Transferable Vote (see, e.g., [1]), the Chamberlin–Courant scheme [6], or the Monroe scheme [15]. The first of these rules proceeds by iterative elimination, while the second and third rule are based on the idea of proportional representation.

However, there are many other settings where the agents need to choose multiple winners, and the voting rules designed for parliamentary elections may not be suitable for them. Consider, for instance, what happens when a university department advertizes several faculty positions: the applicants are the candidates, the members of the hiring committee are the voters, each voter has a ranking of the candidates, but the desiderata for the voting procedure may be quite different from those that arise in the context of voting for the members of parliament. Another example is selecting the recipients of a student fellowship: there is a fixed number of fellowships to be allocated, and the selection committee members rank the candidates, but the fellowship recipients need not "represent" the voters. Yet another example is a movie recommendation system [11], where the aim is to select a set of movies for a prospective user based on the recommendations (i.e., votes) of other users. In fact, the "voters" need not be human: the input to the voting rule may consist of rankings of the candidates according to several different objective criteria, which play the role of voters (consider, e.g., university rankings, where the criteria may include research output, student satisfaction, and external funding). In many of these applications, it is natural to select the winning set by using a simple score-based procedure: each candidate receives a certain number of points from each voter, which is determined by his position in this voter's ranking, and the winners are the top ℓ candidates according to the total score (where ℓ is the number of candidates to be selected). Such voting procedures are easy to implement and understand, and they are often used in practice in scenarios such as the ones described above.

Just as single-winner elections, multi-winner elections are susceptible to *manipulation*: a voter may try to misrepresent his preferences to improve the election outcome from his perspective (we remark that it is not trivial to define "improvement" in the context of multi-winner elections; we discuss this issue in detail in Section 2). For single-winner elections, this problem is known to be pervasive: the classic result of Gibbard and Satterthwaite [12, 19] states that in general, if a voting rule is not susceptible to manipulation, it

Appears in: *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2013), Ito, Jonker, Gini, and Shehory (eds.), May, 6–10, 2013, Saint Paul, Minnesota, USA.*

is trivial in the sense that it does not depend on the agents' preferences or does so in a very limited manner.

Bartholdi et al. [2] suggested that one can try to circumvent this issue by taking an algorithmic perspective, i.e., checking if a successful manipulation can be found in polynomial time. Since then, computational complexity of voting manipulation has been explored for many single-winner voting rules, in a variety of settings (single-voter vs. coalitional manipulation, weighted vs. unweighted votes), see [10, 9] for an overview. Multi-winner elections are susceptible to manipulation, too, and, just like for single-winner elections, it is natural to ask if a manipulative vote is easy to find. Indeed, there are a few papers that consider this problem, both for proportional representation rules [18, 13, 4] and for score-based procedures [14].

Now, somewhat paradoxically, voting rules that are typically used for single-winner elections may fail to output a single winner: e.g., if a rule works by assigning scores to candidates, there may be several candidates with the top score, so the resulting *tie* needs to be broken; a similar issue may arise in the context of multi-winner elections. To the best of our knowledge, all of the existing results on the complexity of manipulating multi-winner voting rules depend on the assumption that ties among the candidates are broken either in the manipulator's favor or adversarially to the manipulator; this is also the assumption made by Bartholdi et al. [2] (and much of the subsequent literature) for single-winner rules. However, this assumption is not as innocuous as it may seem: for single-winner rules, it has recently been shown that the choice of the tie-breaking rule may strongly affect the computational complexity of voting manipulation [17, 16]. One of the goals of this paper is to explore if this is also the case for multi-winner voting rules.

1.1 Our Contribution

We focus on multi-winner voting procedures that are based on scoring rules; this class of rules has been previously considered by Meir et al. [14]. We consider two types of tie-breaking rules: lexicographic rules, which break ties according to a predetermined ordering of the candidates, and a natural randomized rule. For both types of rules, we identify a number of settings where the manipulator's problem is polynomial-time solvable. In particular, we describe efficient algorithms for this problem for three different scenarios:

(1) the underlying voting rule is k-Approval (this result holds for both types of tie-breaking rules);

(2) the number of candidates to be elected is bounded by a constant (this result holds for both types of tie-breaking rules, but the algorithm for the randomized rule is significantly more complicated);

(3) the manipulator's utility function (see Section 2 for the definition of the manipulator's utility) is binary (our algorithm for this scenario only works when ties are broken in the manipulator's favor).

1.2 Related Work

Our work builds on an important paper of Meir et al. [14], who study the problem of manipulating a multi-winner election, for several types of voting rules and under the assumption that ties are broken adversarially to the manipulator. Just as Meir et al., we model the manipulator's preferences by means of a *utility function*, i.e., we assume that the manipulator assigns a numeric utility to each candidate and seeks to maximize his overall utility. The most significant difference between our work and that of Meir et al. is that we investigate a much broader range of tie-breaking rules. Also, Meir et al. mainly focus on approval-like scoring rules, while we consider general scoring rules.

Another line of research that is closely related to our work deals with the role of tie-breaking in single-winner elections. This research agenda was initiated by Obraztsova et al. [17, 16]; some of their results were later extended by Zuckerman and Rosenschein [21]. In particular, our polynomial-time algorithm for manipulating an arbitrary scoring rule under randomized tie-breaking, under the assumption that the number of winners in bounded by a constant, is a direct (though non-trivial) extension of the algorithm for scoring rules that was proposed in [17].

There are several ways of defining a successful manipulation under a multi-winner rule; the utility-based approach is by far not the only one. This issue is discussed by Slinko and White [20] in the context of proportional representation rules, as well as by Meir et al. [14] for score-based rules.

2. PRELIMINARIES

An *election* is given by a set of voters $V = \{v_1, \dots, v_n\}$, a set of candidates $C = \{c_1, \dots, c_m\}$, and a vector $\mathcal{R} = (R_1, \dots, R_n)$, where each R_i, $i = 1, \dots, n$, is a linear order over C; we write $E = (V, C, \mathcal{R})$. R_i is called the *preference order* (or, *vote*) of voter v_i, and the vector $\mathcal{R} = (R_1, \dots, R_n)$ is called a *preference profile*. For readability, we will sometimes denote R_i by $\succ_i$. When $a \succ_i b$ for some $a, b \in C$, we say that voter v_i prefers a to b. We denote by $r(c_j, R_i)$ the *rank* of candidate c_j in the preference order R_i: $r(c_j, R_i) = |\{c \in C \mid c \succ_i c_j\}| + 1$. We denote by $\mathcal{R}_{-i}$ the preference profile that lists the preferences of all voters other than v_i:

$$\mathcal{R}_{-i} = (R_1, \dots, R_{i-1}, R_{i+1}, \dots, R_n).$$

A *voting correspondence* $\mathcal{F}$ is a mapping that, given a preference profile $\mathcal{R}$ over C, selects a non-empty set of *winners* $\mathcal{F}(\mathcal{R}) \subseteq C$. A *voting rule* is a voting correspondence that outputs a unique winner for each preference profile. Voting correspondences can be transformed into voting rules using *tie-breaking rules*. A *tie-breaking rule* T for a voting correspondence $\mathcal{F}$ is a mapping that, given a preference profile $\mathcal{R}$ and the set $S = \mathcal{F}(\mathcal{R})$, outputs a single winner $c \in S$. There are two types of tie-breaking rules that are usually considered in the literature on single-winner elections: *lexicographic tie-breaking rules* and the *uniformly random tie-breaking rule*. A lexicographic tie-breaking rule is given by an ordering $\succ$ over C: given a set $S = \mathcal{F}(\mathcal{R})$, it selects the candidate c such that $c \succ c'$ for all $c' \in S \setminus \{c\}$. The uniformly random tie-breaking rule selects each candidate in $S = \mathcal{F}(\mathcal{R})$ with probability $1/|S|$. For brevity, we will call this tie-breaking rule the *randomized* tie-breaking rule.

Many different types of voting correspondences are used in practice and have been studied in the social choice literature. In this paper, we focus on a class of voting correspondences known as *scoring rules*. A scoring rule for a set of candidates C, $|C| = m$, is specified by a vector of real numbers $\vec{\alpha} = (\alpha_1, \dots, \alpha_m)$ that satisfies $\alpha_1 \geq \dots \geq \alpha_m \geq 0$; we denote the rule that corresponds to a vector $\vec{\alpha}$ by $\mathcal{F}_{\vec{\alpha}}$. Under this rule, a candidate receives α_j points from each voter who ranks him in position j. A candidate's score is his total number of

points, and the winners are the candidates with the highest score. As we are interested in asymptotic complexity results, we will consider *families* of scoring rules, i.e., collections of the form $\mathcal{F} = (\mathcal{F}_{\vec{\alpha}^j})_{j\geq 1}$, where $\vec{\alpha}^j = (\alpha_1^j, \ldots, \alpha_j^j)$. We assume that the entries of each $\vec{\alpha}^j$, $j \geq 1$, are non-negative integers, and that, given j, we can compute $\vec{\alpha}^j$ in polynomial time. Classic examples of (families of) scoring rules include *Borda*, which is given by $\vec{\alpha}^j = (j-1, \ldots, 1, 0)$, *Plurality*, which is given by $\vec{\alpha}^j = (1, 0, \ldots, 0)$, and *k-Approval*, which is given by $\alpha_\ell^j = 1$ if $\ell = 1, \ldots, k$ and $\alpha_\ell^j = 0$ otherwise. For a given $k \geq 1$, we denote the family of scoring rules that corresponds to k-Approval by $\mathcal{F}^k$.

2.1 Voting for a Committee

We consider settings where the voters' goal is to elect not just a single winner, but a *committee* of size ℓ. If we want to use a scoring rule for these purposes, it is natural to choose the ℓ candidates with the highest score. However, we may still need to break ties. Suppose, for instance, that $\ell = 3$ and we have two candidates whose score is 10 and two candidates whose score is 9. Clearly, the candidates whose score is 10 should be elected no matter what, but we need to choose one of the candidates whose score is 9. We will now explain how to do this using a lexicographic or randomized tie-breaking rule. We present our framework for scoring rules; however, it can be used for any voting correspondence that works by assigning scores to candidates and selecting the candidates with the highest score (this class includes such classic voting correspondences as Copeland, Maximin, or Bucklin; see, e.g., [1] for the definitions of these rules).

Fix a scoring rule $\mathcal{F}_{\vec{\alpha}}$ with $\vec{\alpha} = (\alpha_1, \ldots, \alpha_m)$ and an election $E = (V, C, \mathcal{R})$ with $|C| = m$. Given a candidate $c \in C$, let s_c denote c's score in E under $\mathcal{F}_{\vec{\alpha}}$. We say that candidates c and c' are on the same *level* if $s_c = s_{c'}$. There are $p \leq m$ levels, denoted $H_1, \ldots, H_p$; we set $s(H_j)$ to be the score of the candidates in H_j, and assume that $s(H_1) > \cdots > s(H_p)$. Let $W_j = \cup_{q=1}^{j} H_q$. If $|W_j| \leq \ell$, then the tie-breaking rule does not apply to W_j: all candidates in W_j are elected irrespective of the tie-breaking rule. Formally, let $j_0 = \max\{j \mid |W_j| \leq \ell\}$ and set $\mathcal{W} = W_{j_0}$. The set $\mathcal{W}$ is called the *confirmed set*: these are the candidates who will definitely be in the elected committee. The set $\mathcal{P} = H_{j+1}$ is called the *pending set*: these are the candidates to which we must apply the tie-breaking rule. Note that $|H_1| > \ell$ implies $\mathcal{W} = \emptyset$ and $\mathcal{P} = H_1$, and $|\mathcal{W}| = \ell$ implies $\mathcal{P} = \emptyset$. For single-winner elections ($\ell = 1$) we obtain $\mathcal{P} = \emptyset$ if $|H_1| = 1$ and $\mathcal{P} = H_1$ otherwise. Let $\ell' = \ell - |\mathcal{W}|$. The lexicographic tie-breaking rule associated with the ordering $\succ$ selects top ℓ' candidates from the set $\mathcal{P}$ according to $\succ$. The randomized tie-breaking rule selects ℓ' candidates from $\mathcal{P}$ uniformly at random.

2.2 Utilities and Manipulation

The outcomes of a multi-winner election are sets of candidates. Therefore, knowing a voter's preference order is insufficient to determine how this voter would compare two different election outcomes. For instance, if $C = \{a, b, c, d\}$, $\ell = 2$, and v_i's preference order is $a \succ_i b \succ_i c \succ_i d$, it is not clear if v_i would prefer $\{a, d\}$ to $\{b, c\}$. To deal with this difficulty, we assume that for each voter $v_i \in V$ we are given a utility function $u_i : C \to \mathbb{Z}$: $u_i(c)$ is the utility that v_i assigns to candidate c. We assume that u_i is consistent with $\succ_i$, i.e., $c \succ_i c'$ implies $u_i(c) \geq u_i(c')$ for all $c, c' \in C$. Note that this inequality is not strict: we allow a voter to assign the same utility to two distinct candidates even though $\succ_i$ is required to be a strict order. The utility function can be extended to sets of candidates by setting $u_i(S) = \sum_{c\in S} u_i(c)$.

When randomized tie-breaking is employed, the relevant measure of v_i's utility is his *expected utility*, which can be computed as follows. Given an election E and a target committee size ℓ, let $\mathcal{W}$ be the confirmed set, and let $\mathcal{P}$ be the pending set. Let $T_s(\mathcal{P})$ denote the random variable that takes values in the space of all s-subsets of $\mathcal{P}$, with each subset being equally likely. Given a variable ξ, let $\mathbb{E}[\xi]$ denote its expectation. Then v_i's utility in E is

$$u_i(\mathcal{W}) + \mathbb{E}[\sum_{c\in T_{\ell'}(\mathcal{P})} u_i(c)],$$

where $\ell' = \ell - |\mathcal{W}|$.

A voter may want to *manipulate* an election with respect to a given voting rule, i.e., misrepresent his preferences so that the election outcome under this rule improves from his perspective. In single-winner elections the improvement is usually determined with respect to the manipulator's preferences order, i.e., a manipulation is deemed successful if the outcome of non-truthful voting is ranked higher than the outcome of truthful voting in the manipulator's preference order. The computational social choice literature often uses a somewhat different model: it is assumed that the manipulator wants to make a specific candidate p elected, and a manipulation is considered successful if this candidate is the election winner. The advantage of the latter model is that it is easy to extend to voting correspondences: in the *unique winner* variant of the manipulation problem, the manipulator succeeds if p is the unique election winner, whereas in the *co-winner* variant the manipulator succeeds if p is among the election winners.

This model can be further extended to multi-winner elections under lexicographic tie-breaking: one can assume that the manipulator has a preferred committee S, $|S| = \ell$, and the manipulation is successful if and only if the election outcome is exactly S. However, this approach assumes that the manipulator is extremely inflexible, i.e., he will not be satisfied unless the elected committee exactly matches his vision. Further, it is not clear how to interpret it in the context of randomized tie-breaking, where the election outcome is a random variable (this issue arises even for single-winner elections, and is discussed by Obraztsova et al. [16, 17]). Therefore, in our work we choose a different model. Namely, we measure the manipulator's satisfaction by his utility (or, in case of randomized tie-breaking, his expected utility): we ask if the manipulator can vote so that his (expected) utility exceeds a given threshold q; this approach was previously used by Meir et al. [14] in the context of multi-winner elections and by Obraztsova et al. [16, 17] in the context of randomized tie-breaking. We remark that this utility-based framework can be used to model the single-minded manipulator discussed earlier in this paragraph, by assigning a utility of 1 to all candidates in S and a utility of 0 to all other candidates, and setting $q = \ell$.

We are now ready to define the computational problems associated with manipulating a multi-winner election under lexicographic or randomized tie-breaking with respect to a family of scoring rules $\mathcal{F} = (\mathcal{F}_{\vec{\alpha}^j})_{j\geq 1}$; we will refer to these problems as $\mathcal{F}$-LexMultManipulation and $\mathcal{F}$-RandMultManipulation, respectively. To simplify no-

tation, we always assume that the manipulator is the last voter, i.e., v_n, and write $u_n(c) = u(c)$.

An instance of $\mathcal{F}$-LexMultManipulation is given by an election $E = (V, C, \mathcal{R})$ with $|C| = m$, a committee size ℓ, a preference order $\succ$, the manipulator's utility function $u : C \to \mathbb{Z}$, which satisfies $u(c) \geq u(c')$ if and only if $c \succ_n c'$, and a non-negative rational number q. It is a "yes"-instance if there exists a vote L such that the manipulator's utility in the election $(V, C, (\mathcal{R}_{-n}, L))$ is at least q, assuming that the winning committee is selected by applying $\mathcal{F}$ and breaking ties lexicographically according to $\succ$. Otherwise, it is a "no"-instance.

Similarly, an instance of $\mathcal{F}$-RandMultManipulation is given by an election $E = (V, C, \mathcal{R})$ with $|C| = m$, a committee size ℓ, the manipulator's utility function $u : C \to \mathbb{Z}$, which satisfies $u(c) \geq u(c')$ if and only if $c \succ_n c'$, and a non-negative rational number q. It is a "yes"-instance if there exists a vote L such that the manipulator's expected utility in $(V, C, (\mathcal{R}_{-n}, L))$ is at least q, assuming that the winning committee is selected by applying $\mathcal{F}$ and using the randomized tie-breaking rule. Otherwise, it is a "no"-instance.

3. K-APPROVAL

For k-Approval, we can efficiently solve the manipulator's problem under both lexicographic and randomized tie-breaking.

Theorem 3.1. *The problems $\mathcal{F}^k$-LexMultManipulation and $\mathcal{F}^k$-RandMultManipulation can be decided in polynomial time for every $k \geq 1$.*

Proof. Fix a $k \geq 1$. Consider a $(n-1)$-voter election $E' = (V \setminus \{v_n\}, C, \mathcal{R}_{-n})$. Let $\mathcal{P}'$ and $\mathcal{W}'$ be, respectively, the pending set and the confirmed set in E'. Set $\mathcal{X}' = C \setminus (\mathcal{P}' \cup \mathcal{W}')$. Let s^+ be the lowest k-Approval score among the candidates in $\mathcal{W}'$ (set $s^+ = +\infty$ if $\mathcal{W}' = \emptyset$), let s^- be the highest k-Approval score among the candidates in $\mathcal{X}'$ (set $s^- = -\infty$ if $\mathcal{X}' = \emptyset$), and let s be the k-Approval score of the candidates in $\mathcal{P}'$ (if $\mathcal{P}' = \emptyset$, s remains undefined). Let $\mathcal{W}'_- \subseteq \mathcal{W}'$ be the set of candidates whose k-Approval score is s^+, and let $\mathcal{X}'_+ \subseteq \mathcal{X}'$ be the set of candidates whose k-Approval score is s^-; also, set $\mathcal{W}'_+ = \mathcal{W}' \setminus \mathcal{W}'_-$ and $\mathcal{X}'_- = \mathcal{X}' \setminus \mathcal{X}'_+$. Note that $s^- < s^+$, and if $\mathcal{P}' \neq \emptyset$, we have $s^- < s < s^+$.

Let E be the election obtained after the manipulator votes, and suppose that in E the confirmed set is $\mathcal{W}$ and the pending set is $\mathcal{P}$; also, set $\mathcal{X} = C \setminus (\mathcal{W} \cup \mathcal{P})$. We will now argue that, no matter how the manipulator votes, we have $\mathcal{W}'_+ \subseteq \mathcal{W}$ and $\mathcal{X}'_- \subseteq \mathcal{X}$, i.e., points allocated to candidates in $\mathcal{W}'_+ \cup \mathcal{X}'_-$ do not affect the election outcome. Indeed, in E the score of every candidate in $\mathcal{W}'_+$ will be at least $s^+ + 1$, and there can be at most $|\mathcal{W}'| \leq \ell$ candidates with such score, so every candidate in $\mathcal{W}'_+$ will end up in $\mathcal{W}$. Further, in E the score of every candidate in $\mathcal{X}'_-$ will be at most s^-, and there are at least $|\mathcal{P}'| + |\mathcal{W}'| \geq \ell$ candidates whose score is at least $s^- + 1$, so the score of s^- will be insufficient for being placed in $\mathcal{P}$. This observation is useful, as it allows us to "dump" extra points by assigning them to $\mathcal{X}'_- \cup \mathcal{W}'_+$.

Now, suppose that the manipulator has decided to approve k_w candidates in $\mathcal{W}'_-$. Then, to maximize his utility, he has to approve k_w candidates in $\mathcal{W}'_-$ with the highest utility. A similar argument works for $\mathcal{P}'$ and $\mathcal{X}'_+$. As for the candidates in $\mathcal{W}'_+ \cup \mathcal{X}'_-$, it does not matter which ones he chooses to approve, since, as argued above, his vote will not change the status of these candidates. Thus, the outcome of the election is completely determined by a triple of non-negative integers (k_w, k_p, k_x), where k_w, k_p, and k_x are, respectively, the number of candidates in $\mathcal{W}'_-$, $\mathcal{P}'$, and $\mathcal{X}'_+$ that the manipulator approves. Hence, the manipulator can go over all triples of integers $(k_w, k_p, k_x) \in \{0, \dots, k\}^3$, and, for each triple, check if it corresponds to a valid vote and compute the (expected) utility that he obtains from approving k_w highest-utility candidates from $\mathcal{W}'_-$, k_p highest-utility candidates from $\mathcal{P}'$, and k_x highest-utility candidates from $\mathcal{X}'_+$, and distributing the remaining points (if any) among the rest of the candidates. The manipulator can then check if the expected utility from the best such triple is at least q. Clearly, (k_w, k_p, k_x) corresponds to a valid vote if and only if

- $0 \leq k_w \leq |\mathcal{W}'_-|$,
- $0 \leq k_p \leq |\mathcal{P}'|$,
- $0 \leq k_x \leq |\mathcal{X}'_+|$, and
- $0 \leq k - k_w - k_p - k_x \leq |\mathcal{X}'_-| + |\mathcal{W}'_+|$,

and the manipulator's (expected) utility from any such vote can be computed in time $O(k)$. Thus, the overall running time of our algorithm is $O(k^4)$. Since we can assume that $k \leq m$, this running time is polynomial in the input size. □

By refining the analysis in the proof of Theorem 3.1, we can prove that both $\mathcal{F}^k$-LexMultManipulation and $\mathcal{F}^k$-RandMultManipulation can be solved in time $\mathcal{O}(m^2)$. We will now provide a brief overview of the $\mathcal{O}(m^2)$ algorithm, omitting details due to space constraints.

We use the same notation as in the proof of Theorem 3.1. Abusing notation, we denote by $u(\mathcal{R})$ the manipulator's (expected) utility given the preference profile $\mathcal{R}$.

Suppose first that $\mathcal{P}' \neq \emptyset$. The key observation is that any vote L submitted by v_n must fall into one of the following four cases.

Case 1: The pending set $\mathcal{P}$ is empty after L is submitted; let us call this set $\mathcal{L}_1$. We observe that one can easily find a vote L_1 such that $L_1 = \arg\max_{L \in \mathcal{L}_1} u(\mathcal{R}_{-n}, L)$. This can be done simply by assigning $\ell - |\mathcal{W}'|$ approval points to v_n's favorite candidates in $\mathcal{P}'$, and assigning the rest of the approval points to $\mathcal{X}' \cup \mathcal{W}'$ arbitrarily. If this can be done, then $\mathcal{L}_1 \neq \emptyset$, and L_1 exists.

Case 2: $\emptyset \neq \mathcal{P} \subseteq \mathcal{P}'$, and the score of each candidate in $\mathcal{P}$ is $s+1 < s^+$. We denote the set of votes of this type by $\mathcal{L}_2$; again, one can find an $L_2 = \arg\max_{L \in \mathcal{L}_2} u(\mathcal{R}_{-n}, L)$ by trying, for all $r = \ell - |\mathcal{W}'| + 1, \dots, k$, to assign a point to r of the manipulator's favorite candidates in $\mathcal{P}'$ and "dumping" the $k - r$ remaining points on $\mathcal{X}' \cup \mathcal{W}'$. The utility from this assignment should be computed for all r, and L_2 corresponds to a choice of r that maximizes this utility.

Case 3: The score of each candidate in $\mathcal{P}$ is s^+. This means that $s = s^+ - 1$ and that some of the candidates in $\mathcal{P}'$ were approved by v_n, and together with $\mathcal{W}'_-$ form the pending set $\mathcal{P}$. For the set of votes satisfying these conditions, $\mathcal{L}_3$, we have $\mathcal{W} \supseteq \mathcal{W}'_+$. In this case, finding an optimal vote L_3 requires going over all possible approvals of the top k_p candidates from $\mathcal{P}'$ (to be moved

to $\mathcal{P}$) and k_w candidates from $\mathcal{W}'_-$ (to be moved from $\mathcal{P}$ to $\mathcal{W}$). In contrast with the method for finding L_1 and L_2, finding L_3 requires $\mathcal{O}(m^2)$ time.

Case 4: The score of each candidate in $\mathcal{P}$ after v_n votes is s. We call the set of these votes $\mathcal{L}_4$. This can occur only if a sufficiently small number of candidates from $\mathcal{P}'$ were approved (so as not to be in Case 3), and if any candidates from $\mathcal{X}'_+$ are now in $\mathcal{P}$, then $s_- = s - 1$. In order to maintain the score of $\mathcal{P}$ at s, we must have that $\mathcal{P}' \neq \emptyset$; moreover, we must consider the case where $\mathcal{X}'_+$ has a score of $s-1$ and the case where it is strictly less than $s-1$. In the former case, it is possible to make some of the candidates in $\mathcal{X}'_+$ part of $\mathcal{P}$. In the latter case, no candidate in $\mathcal{X}'$ can be part of $\mathcal{P}$. In either case, an optimal vote can be found in $\mathcal{O}(m^2)$ time.

Finally, having computed $L_1, \ldots, L_4$, we can find an optimal manipulation by picking the best of these four votes, i.e., setting $L^* = \arg\max_{i=1,\ldots,4} u(\mathcal{R}_{-n}, L_i)$.

The case $\mathcal{P} = \emptyset$ can be analyzed similarly, except that we only have two cases to consider. This completes the description of our $\mathcal{O}(m^2)$ algorithm.

4. SMALL COMMITTEES

In this section we consider the setting where the committee size ℓ is bounded by a constant. Our first result for this case is a polynomial-time algorithm for finding a successful manipulation, under the assumption that ties are broken lexicographically.

THEOREM 4.1. *For any family of scoring rules $\mathcal{F}$ the problem $\mathcal{F}$-LEXMULTMANIPULATION admits an algorithm that runs in time $\mathcal{O}(nm^{\ell+2})$.*

PROOF. Fix the non-manipulators' preference profile $\mathcal{R}_{-n}$. We will go over all committees of size ℓ; for each such committee D we will check if the manipulator v_n can vote so that the election outcome is D. We will then choose the highest-utility committee for which the answer is positive. It remains to show how to determine this for a specific committee D.

Fix a committee $D = \{d_1, \ldots, d_\ell\} \subseteq C$. First, observe that if v_n can vote so that the election outcome is D, then he has a manipulative vote such that the election outcome is D and the candidates in D are placed in the top ℓ positions in his vote. We will go over all possible ways of ordering the candidates in D in the top ℓ positions in the vote.

Fix one such ordering π, and assume without loss of generality that π places d_i in position i for $i = 1, \ldots, \ell$. We will use a greedy algorithm to fill the bottom $m - \ell$ positions. That is, we order the candidates in $C \setminus D$ according to the number of points they receive from the first $n-1$ voters (from the smallest to the largest); if c and c' have the same score and our lexicographic tie-breaking rule favors c over c', then c' appears before c in our ordering. We then place these candidates in positions $\ell+1, \ldots, m$ in v_n's vote according to our ordering.

We claim that π can be completed to a successful manipulation if and only if the resulting vote makes D the winning set. The "if" direction is obvious. For the "only if" direction, suppose that there is some assignment of voters in $C \setminus D$ to the last $m - \ell$ slots that makes D the winning set, and let $\mathcal{L}$ denote the set of all such assignments. Let us denote by L_G the greedy assignment described above, and let L_M be an assignment in $\mathcal{L}$ that shares the longest common prefix with L_G.

Suppose for the sake of contradiction that $L_G \neq L_M$, and let j be the first position where L_G and L_M differ. Suppose that the j-th position contains candidate c under L_G and candidate d under L_M. Then d can be placed in position j without becoming a winner. Since the greedy method did not place d in positions $1, \ldots, j$, we conclude that d either has a higher score than c or is prefered to c by the tie-breaking rule. Thus, c, too can be placed in position j without becoming a winner. On the other hand, moving d from position j to a position j' with $j' > j$ would not make him a winner either. This means that we can swap c and d in L_M and still obtain an assignment in $\mathcal{L}$. Moreover, the resulting assignment shares a strictly longer prefix with L_G, a contradiction with the definition of L_M.

For each committee D, the score of each candidate in $C \setminus D$ can be computed in time $\mathcal{O}(n)$; sorting the candidates according to these scores takes $\mathcal{O}(m \log m)$ steps. For each committee D and each way of ordering the candidates in D in the top ℓ positions, our greedy algorithm runs in linear time. There are $\binom{m}{\ell}$ committees to consider; for each committee, there are $\ell!$ ways to order its members. Thus, the running time of our algorithm is $\mathcal{O}(\binom{m}{\ell}(\ell! + nm \log m)) = \mathcal{O}(nm^{\ell+2})$. □

As shown by Obraztsova et al. [16, 17], manipulating single-winner elections when the randomized tie-breaking rule is used may require techniques that are fundamentally different from those that work for lexicographic tie-breaking. It turns out that this is also the case for multi-winner elections. Indeed, our next result is a direct analogue of Theorem 4.1; however, its proof is substantially more complicated than that of Theorem 4.1.

THEOREM 4.2. *For any family of scoring rules $\mathcal{F}$ the problem $\mathcal{F}$-RANDMULTMANIPULATION is in* P *when ℓ is bounded by a constant.*

PROOF. Consider an election $(V, C, \mathcal{R})$ with $|C| = m$ and an m-candidate scoring rule $\mathcal{F}_{\vec{\alpha}} \in \mathcal{F}$ with a scoring vector $\vec{\alpha} = (\alpha_1, \ldots, \alpha_m)$, Let $\mathbf{s} = (s_1, \ldots, s_m)$ be the vector of the candidates' scores in $(V, C, \mathcal{R}_{-n})$. For each $k \leq \ell$ and each subset $\mathcal{W}_k \subseteq C$ of size k, we check if the manipulator can vote so that the confirmed set is $\mathcal{W}_k$. If this is indeed the case, we find the best set of $\ell - k$ pending winners for this choice of $\mathcal{W}_k$; that is, we identify a set $\mathcal{P}_k$ with $|\mathcal{P}_k| > \ell - k$ such that after the manipulator's vote the confirmed set is $\mathcal{W}_k$, the (identical) scores of the candidates in $\mathcal{P}_k$ are strictly less than those of any $c \in \mathcal{W}_k$, and the manipulator's expected utility from $\mathcal{P}_k$ is maximized. Notice that the requirement $|\mathcal{P}_k| > \ell - k$ is necessary; otherwise, $\mathcal{P}_k \cup \mathcal{W}_k$ are the confirmed winners, which contradicts our objective of having $\mathcal{W}_k$ as the confirmed winners. We then compute the manipulator's expected utility from having the candidates in $\mathcal{W}_k$ as the confirmed winners and the candidates in $\mathcal{P}_k$ as the pending winners, and select a triple $(k, \mathcal{W}_k, \mathcal{P}_k)$ that maximizes the manipulator's expected utility.

The candidate set C has at most $\sum_{k=1}^{\ell} \binom{m}{k} \in \mathcal{O}(m^\ell)$ subsets of size at most ℓ; thus, it remains to show that for each subset of size at most ℓ the procedure described in the previous paragraph can be implemented in polynomial time. Fix

a $k \leq \ell$ and a set $\mathcal{W}_k$. First, we pick k entries of $\vec{\alpha}$; these are the scores that we will assign to candidates in $\mathcal{W}_k$. There are $\binom{m}{k} = \mathcal{O}(m^\ell)$ ways of choosing such a set of scores; we go over all possible choices. We then order the candidates in $\mathcal{W}_k$ by decreasing order of scores under $\mathbf{s}$, and assign the lowest among the selected k scores to the first candidate, the second lowest to the second candidate and so on. If $\mathcal{W}_k$ can be made confirmed winners under some assignment of the k scores selected, then in particular they can be made confirmed winners under this assignment. Now, let $H_1, \ldots, H_p$ be the levels of the candidates in $C \setminus \mathcal{W}_k$. We renumber the candidates in $C \setminus \mathcal{W}_k$ so that for all $i \in 1, \ldots, p-1$, all candidates in H_i are before the candidates in H_{i+1}. Given a level H_i, we order the candidates in H_i so that if $c, c' \in H_i$ and the manipulator prefers c to c', then c' precedes c. Let $\vec{\alpha}' = \{\alpha_{i_1}, \ldots, \alpha_{i_{m-k}}\}$ be the remaining $m-k$ scores that the manipulator needs to assign; we assume $\alpha_{i_1} \leq \cdots \leq \alpha_{i_{m-k}}$.

We assign $\alpha_{i_1}, \ldots, \alpha_{i_{|H_1|}}$ to H_1 in that order. Similarly, we assign $\alpha_{i_{|H_1|+1}}, \ldots, \alpha_{i_{|H_1|+|H_2|}}$ to H_2 and so on until all scores are assigned. This assignment, denoted σ_0, ensures that at each level, the manipulator's favorite candidates from that level receive the highest scores. Let Φ be the highest score of any candidate in $C \setminus \mathcal{W}_k$ under σ_0. Observe that for every score assignment to candidates in $C \setminus \mathcal{W}_k$ the score of some candidate in $C \setminus \mathcal{W}_k$ after the manipulator's vote is at least Φ. Thus, if Φ is greater than or equal to the score of some $c \in \mathcal{W}_k$, then $\mathcal{W}_k$ cannot be made confirmed winners using this score assignment, and we proceed to check a different assignment of scores to $\mathcal{W}_k$. Therefore, from now one we assume that the score of each candidate in $\mathcal{W}_k$ is greater than Φ.

Let $\mathcal{P}_0$ be the set of candidates whose score is Φ after submitting σ_0. We can try to modify σ_0 in order to increase the manipulator's utility, by swapping some candidates in the vote. We claim that reassigning scores given to members of $\mathcal{P}_0$ will either result in a non-tied outcome, or decrease the manipulator's expected utility from the set of tied candidates. Indeed, suppose that a candidate $c \in \mathcal{P}_0$ received a score of β and now receives a higher score β'; this increases his score to be strictly more than Φ. If this results in a strictly higher utility for the manipulator, this means that the manipulator can strictly increase his utility by greedily assigning the highest scores in $\vec{\alpha}'$ to the candidates he prefers the most, with no ties formed. On the other hand, if we assign a lower score to c, this means that some other candidate in a higher level receives a higher score, and the same argument applies. Thus, any swap we make will only involve candidates not in $\mathcal{P}_0$. However, note that the manipulator's utility is unaffected by candidates whose score is less than Φ. Thus, for any candidate c not in $\mathcal{P}_0$, we can just check if there is some score that will give him a total score of Φ. If such $c \in (C \setminus \mathcal{W}_k) \setminus \mathcal{P}_0$ exists, and adding c to $\mathcal{P}_0$ increases the manipulator's expected utility, we can add c to $\mathcal{P}_0$. Having done so for each candidate, we denote the resulting set by $\mathcal{P}_1$.

We claim that $\mathcal{P}_1$ is indeed the set of pending candidates we require. To show this, we will now argue that if $|\mathcal{P}_1| \leq \ell - k$ then one of the following two cases holds:

Case 1: Given $\mathcal{W}_k$ and the scores we assign to members of $\mathcal{W}_k$, it is impossible to find a score assignment such that $\mathcal{W}_k$ are confirmed winners.

Case 2: Even if there is a set $\mathcal{P}_k$ of pending winners, there is a set $\mathcal{P}'$ of candidates of size exactly $\ell - k$ such that the manipulator's utility from $\mathcal{W}_k \cup \mathcal{P}'$ is at least his expected utility from having $\mathcal{W}$ as the confirmed winners and $\mathcal{P}_k$ as the pending winners.

Observe that both cases imply that if $|\mathcal{P}_1| \leq \ell - k$ we can just move on to another score assignment to $\mathcal{W}_k$, and ignore the current assignment: it is either impossible to have $\mathcal{W}_k$ as the confirmed winners, or there is another candidate set with the same utility that can be made confirmed winners and will be found in some other iteration. We must show that indeed one of these two cases holds.

If neither case holds, there exists a vote σ' such that if the manipulator submits σ', the set of confirmed winners is $\mathcal{W}_k$, the set of pending winners is $\mathcal{P}_k$, and for any set $\mathcal{P}' \subseteq C \setminus \mathcal{W}_k$ such that $|\mathcal{P}'| = \ell - k$ and the set $\mathcal{W}_k \cup \mathcal{P}'$ is a feasible set of winners it holds that the manipulator's expected utility from having $\mathcal{W}$ as the confirmed winners and $\mathcal{P}_k$ as the pending winners is greater than his utility from $\mathcal{W}_k \cup \mathcal{P}'$.

First, consider the case where both confirmed and pending candidates receive more than Φ points. Let $c_{j_1}, \ldots, c_{j_{\ell-k}}$ be the manipulator's most preferred $\ell - k$ candidates in $\mathcal{P}_k$; by assumption, we must have that the manipulator's expected utility from $\mathcal{P}'$ is at most $\sum_{p=1}^{\ell-k} u(c_{j_p})$. Let $\mathcal{S}$ be the set consisting of these $\ell - k$ candidates and $\mathcal{W}_k$. Consider any candidate $c_j \in \mathcal{S}$ and suppose the manipulator grants $\alpha_{j'}$ points to c_j. The score of c_j after the manipulator votes is strictly more than Φ; thus $j' < j$. We set $\mathcal{S}' = \{c_{j'} \in C \mid \alpha_{j'} \text{ is assigned to some } c_j \text{ under } \sigma'\}$.

Now, consider the vote obtained from σ_0 by swapping the votes given to c_j and its corresponding candidate $c_{j'}$. Observe that some candidates can be in two such swaps—once acting as c_j and once as $c_{j'}$—in this case we begin from the swap which uses the candidate as c_j and afterwards we use the candidate who was put into his place for the next swap. All candidates in $C \setminus \mathcal{S} \setminus \mathcal{S}'$ do not have their scores changed, so they still get at most Φ points; more importantly, all candidates in $\mathcal{S}$ now get strictly more than Φ points. Further, all candidates in $\mathcal{S}' \setminus \mathcal{S}$ get less than Φ points. Thus, in this case $\mathcal{S}$ are the confirmed winners and the manipulator's expected utility is at least as high as that from having $\mathcal{W}_k$ as the confirmed winners and $\mathcal{P}_k$ as the pending winners, a contradiction. The other case is when the candidates in $\mathcal{W}_k$ have more than Φ points, but the candidates in $\mathcal{P}_k$ have exactly Φ points. This case is handled similarly; we omit the details to avoid repetition. □

5. BINARY UTILITIES

We will now show that if the manipulator's utilities are binary, i.e., $u(c) \in \{0, 1\}$ for all $c \in C$, then for any family of scoring rules $\mathcal{F}$ the problem $\mathcal{F}$-LexMultManipulation is in P, assuming that ties are broken in the manipulator's favor. Binary utilities imply that a voter who wishes to manipulate the election has a considerably easier problem: he needs to maximize the number of desirable candidates in the committee. We will argue that this can be done in polynomial time when ties are broken in the manipulator's favor. We remark that Meir et al. [14] prove a similar result for adversarial tie-breaking; however, our argument is different from the one presented in [14].

Theorem 5.1. *Fix a family of scoring rules $\mathcal{F}$. If $u(c) \in \{0, 1\}$ for all $c \in C$ and the ordering $\succ$ used by the lexi-*

cographic tie-breaking rule coincides with R_n, then $\mathcal{F}$-LexMultManipulation *is in* P.

Proof. Consider a candidate set C, let $m = |C|$, and let $\mathcal{F}_{\vec{\alpha}}$, $\vec{\alpha} = (\alpha_1, \ldots, \alpha_m)$, be the m-candidate scoring rule in $\mathcal{F}$. We divide the candidate set into $G = \{c \in C \mid u(c) = 1\}$ and $B = \{c \in C \mid u(c) = 0\}$. First we note that it is never beneficial for the manipulator to rank a member of B above a member of G. Thus, v_n assigns the $|G|$ highest scores to G and the $|B|$ lowest scores to B. Note that this means that v_n votes truthfully, in the sense that he never ranks a candidate with utility 0 above a candidate with utility 1. We sort B in decreasing order according to score. If $c, c' \in B$ have the same score, we arrange them so that the candidate who is ranked lower by $\succ$ is first. Set $|B| = k$; we assign α_m to the first candidate in B according to our reordering, α_{m-1} to the second and so on, until the last candidate in B gets the highest score α_{m-k+1}. Let t be the maximum score of a candidate in B after the manipulator assigned $\alpha_{m-k+1}, \ldots, \alpha_m$ to B. Then t is the minimum score any candidate in B will get under any scoring rule. It is the "score to beat" for the candidates in G; any candidate in G who can get at least t points will be either a confirmed winner or a pending winner. Following Obraztsova et al. [17] we check all possible winning scores achievable by B that are greater than t; as observed in [17], there are polynomially many such scores. We denote the set of all possible winning scores achievable by B by $W(B)$.

Now, given such a score $t' \in W(B)$, we check what is the best that the manipulator can do. Observe that since ties are always broken in G's favor, all of G's members benefit v_n equally. We sort G according to the score. Let $H_1, \ldots, H_p$ be the levels of G's members, where s_j is the score of H_j. We assume that $s_1 < \cdots < s_p$. Given a score s_j and t', we set $A_j = \{\alpha_\ell \in \vec{\alpha} \mid s_j + \alpha_\ell \geq t'\}$; A_j is the set of all scores for which the members of H_j can be made winners, given t'.

We now proceed as follows: We assign members of H_1 the scores in A_1; if $|H_1| > |A_1|$, we arbitrarily assign $|A_1|$ scores to $|A_1|$ members of H_1. If $|H_1| \leq |A_1|$, we assign the $|H_1|$ highest scores in A_1 to the members of H_1. We denote the scores assigned to H_1 by Z_1. We repeat the process for H_2, but only assign $A_2 \setminus Z_1$ in a similar manner, and denote the set of assigned scores to H_1 and H_2 by Z_2. In general, let Z_j be the set of assigned scores to $H_1, \ldots, H_j$, we assign the scores $A_{j+1} \setminus Z_j$ to H_{j+1} in the same manner and set Z_{j+1} to be the set of assigned scores.

This process terminates with the maximum number of candidates in G getting a score of at least t'. Indeed, let us denote our allocation Greedy and consider some optimal score allocation Opt. We denote by Greedy_j the number of candidates in levels $H_1, \ldots, H_j$ who were given a score of at least t' under Greedy, and we similarly define Opt_j.

Lemma 5.2. *For any j we have* $\text{Greedy}_j \geq \text{Opt}_j$.

Proof. We proceed by induction on j. For $j = 1$, the number of candidates that are given a score of more than t' cannot exceed $\min\{|A_1|, |H_1|\}$ under any vote by v_n. Since Greedy assigns at least $|A_1|$ candidates a score of at least t', we are done. We now assume that the statement of the lemma is true for $j - 1$ and show that it holds for j. Again, note that the number of candidates in H_j who get a score of more than t' is no more than $\min\{|A_j|, |H_j|\}$. Greedy assigns less than $\min\{|A_j|, |H_j|\}$ candidates a winning score only if some scores from A_j were assigned to $\bigcup_{r=1}^{j-1} H_r$. Similarly, if H_j has more winners under Opt than under Greedy, it must be because some of the scores in A_j that were assigned by Greedy to members of $\bigcup_{r=1}^{j-1} H_r$ are now assigned to members of H_j. Let us observe the scores that Opt did assign to $H_1, \ldots, H_{j-1}$; we note that if we assign the weights used by Opt according to the Greedy method, we will get by the induction hypothesis a better result for $H_1, \ldots H_{j-1}$. Moreover, observe that every score that Opt has assigned to H_j instead of some H_r with $r < j$, can be assigned to a candidate in $H_1, \ldots, H_{j-1}$ according to Greedy; indeed, if that score was not assigned by Greedy to H_j, and could make a member of H_j winning, it must have been assigned to some member of $H_1, \ldots, H_{j-1}$, otherwise it means all members of H_j have been assigned scores to push them over a score of t'. Thus, every winning candidate in H_j under Opt that was not winning under Greedy can be matched to a candidate that is winning under Greedy, but not under Opt. This implies that for j we have $\text{Greedy}_j \geq \text{Opt}_j$. □

Lemma 5.2 implies that the number of candidates in G that receive a score of at least t' is maximized by using Greedy. Note that we do not consider ties in this scenario; candidates in G who have a score of exactly t' will always be chosen over candidates in B by our assumption on the tie-breaking rule. Also, since the manipulator assigns the same utility to all candidates in G, we only need to maximize the number of candidates in G with a score of more than t', which can be achieved by using Greedy according to Lemma 5.2.

After calculating the maximum utility that the manipulator can get for every minimum winning score $t' \in W(B)$, we choose a vote which guarantees the manipulator the maximum utility; this is the optimal vote for v_n. □

6. CONCLUSIONS AND FUTURE WORK

We have explored some of the computational aspects of manipulation in multi-winner elections, with a particular focus on the role of tie-breaking rules. Our work can be viewed as a merge of two recent lines of enquiry: the complexity of manipulation in multi-winner elections [14] (under adversarial tie-breaking, i.e., in the unique-winner model) and the complexity of manipulation in single-winner elections under lexicographic and randomized tie-breaking [16, 17]. Our results show that the manner in which ties are broken plays an important role in the analysis of voting manipulation.

The most obvious open problem suggested by our work is determining the exact complexity of finding a manipulative vote under general scoring rules when the committee size is not bounded by a constant and the manipulator's utility function can be arbitrary. We conjecture that this problem is NP-hard.

Further, we have shown that the manipulation problem is in P if the committee size is bounded by a constant, both for lexicographic and for randomized tie-breaking (see Theorems 4.1 and 4.2, respectively). However, the running time of our algorithms for small committees is of the form $\Omega(m^\ell)$, where ℓ is the target committee size and m is the number of candidates. Thus, while our results place the computational problems considered in this paper in the complexity class XP with respect to ℓ, they *do not* imply that these problems are *fixed-parameter tractable*, i.e., belong to the complexity class FPT (see, e.g., [7] for the definitions of these complex-

ity classes). It would be interesting to verify if this is indeed the case.

One can also explore the impact of tie-breaking rules on the complexity of other forms of dishonest behavior in multi-winner elections, such as control [3] or bribery [8]; while the complexity of control in multi-winner elections has been considered by Meir et al. [14] (under adversarial tie-breaking), to the best of our knowledge, there has been no work on bribery in such elections.

We remark that we have briefly discussed why we chose to model the manipulator's preferences by an additive utility function. However, in the context of committee elections voters may have preferences that are not additive. For instance, a voter may strongly believe that the set of winners should include at least one candidate from the set $\{a, b\}$ as well as at least one candidate from the set $\{c, d\}$; it can be shown that these preferences cannot be captured by an additive utility function. To model such scenarios, we need a framework that enables us to succinctly represent preferences over entire committees and to reason about such preferences. While there is a significant body of research on representing preferences over sets of alternatives (see, e.g., [5]), to the best of our knowledge the problem of manipulation with respect to such preferences has not been analyzed in the literature.

Acknowledgments Svetlana Obraztsova was supported by the project ALGONOW of the research funding program THALIS (co-financed by the European Social Fund-ESF and Greek national funds) as well as by grants of Russian goverment (FCP and RFFI). Yair Zick was supported by SINGA graduate fellowship. Edith Elkind was supported by National Research Foundation (Singapore) under grant 2009-08.

7. REFERENCES

[1] K. Arrow, A. Sen, and K. Suzumura, editors. *Handbook of Social Choice and Welfare, Volume 1*. Elsevier, 2002.

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

[3] J. Bartholdi, C. Tovey, and M. Trick. How hard is it to control an election? *Mathematical and Computer Modeling*, 16(8/9):27–40, 1992.

[4] N. Betzler, A. Slinko, and J. Uhlmann. On the computation of fully proportional representation. Technical Report 1952497, SSRN, 2011.

[5] R. I. Brafman, C. Domshlak, S. E. Shimony, and Y. Silver. Preferences over sets. In *Twenty First National Conference on Artificial Intelligence (AAAI 06)*, pages 1101–1106, 2006.

[6] J. Chamberlin and P. Courant. Representative deliberations and representative decisions: Proportional representation and the Borda rule. *American Political Science Review*, 77(3):718–733, 1983.

[7] R. Downey and M. Fellows. *Parameterized Complexity*. Springer-Verlag, 1999.

[8] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. How hard is bribery in elections? *Journal of Artificial Intelligence Research*, 35:485–532, 2009.

[9] P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. Using complexity to protect elections. *Communications of the ACM*, 53(11):74–82, 2010.

[10] P. Faliszewski and A. Procaccia. AI's war on manipulation: Are we winning? *AI Magazine*, 31(4):53–64, 2010.

[11] S. Ghosh, M. Mundhe, K. Hernandez, and S. Sen. Voting for movies: The anatomy of recommender systems. In *Third Annual Conference on Autonomous Agents*, pages 434–435, 1999.

[12] A. Gibbard. Manipulation of voting schemes: A general result. *Econometrica*, 41(4):pp. 587–601, 1973.

[13] T. Lu and C. Boutilier. Budgeted social choice: From consensus to personalized decision making. In *Twenty Second International Joint Conference on Artificial Intelligence (IJCAI 2011)*, pages 280–286, 2011.

[14] R. Meir, A. D. Procaccia, J. S. Rosenschein, and A. Zohar. Complexity of strategic behavior in multi-winner elections. *Journal of Artificial Intelligence Research*, 33:149–178, 2008.

[15] B. Monroe. Fully proportional representation. *American Political Science Review*, 89(4):925–940, 1995.

[16] S. Obraztsova and E. Elkind. On the complexity of voting manipulation under randomized tie-breaking. In *Twenty Second International Joint Conference on Artificial Intelligence (IJCAI 2011)*, pages 319–324, 2011.

[17] S. Obraztsova, E. Elkind, and N. Hazon. Ties matter: Complexity of voting manipulation revisited. In *Tenth International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, pages 71–78, 2011.

[18] A. Procaccia, J. Rosenschein, and A. Zohar. On the complexity of achieving proportional representation. *Social Choice and Welfare*, 30(3):353–362, 2008.

[19] M. Satterthwaite. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10(2):187–217, 1975.

[20] A. Slinko and S. White. Proportional representation and strategic voters. *Journal of Theoretical Politics*, 22(3):301–332, 2010.

[21] M. Zuckerman and J. Rosenschein. Manipulation with randomized tie-breaking under Maximin (extended abstract). In *Eleventh International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, pages 1315–1317, 2012.