Complexity of Safe Strategic Voting

Noam Hazon
Department of Computer Science,
Bar Ilan University, Israel
hazonn@cs.biu.ac.il

Edith Elkind
School of Physical and Mathematical Sciences,
Nanyang Technological University, Singapore
eelkind@gmail.com

ABSTRACT

We investigate the computational aspects of *safe manipulation*, a new model of coalitional manipulation that was recently put forward by Slinko and White [11]. In this model, a potential manipulator v announces how he intends to vote, and some of the other voters whose preferences coincide with those of v may follow suit. Depending on the number of followers, the outcome could be better or worse for v than the outcome of truthful voting. A manipulative vote is called *safe* if for some number of followers it improves the outcome from v's perspective, and can never lead to a worse outcome. In this paper, we study the complexity of finding a safe manipulative vote for a number of common voting rules, including Plurality, Borda, k-approval, and Bucklin. We provide both algorithms and hardness results, demonstrating that the complexity of our problem depends both on the voters' weights and the structural properties of the tie-breaking rule. We also propose two ways to extend the notion of safe manipulation to the setting where the followers' preferences may differ from those of the leader, and study the computational properties of the resulting extensions.

Categories and Subject Descriptors

I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems; I.2.4 [**Knowledge representation formalisms and methods**]

General Terms

Theory

Keywords

Voting, Coalitional manipulation, Computational complexity

1. INTRODUCTION

Computational aspects of voting, and, in particular, voting manipulation, have recently received a lot of attention in the multiagent literature. While the complexity of the manipulation problem for a single voter is quite well understood (specifically, this problem is known to be efficiently solvable for most common voting rules with the notable exception of STV [1, 2]), the more recent work has mostly focused on coalitional manipulation, i.e., manipulation by multiple, possibly weighted voters. In contrast to the single-voter case, coalitional manipulation tends to be hard. Indeed, it has been shown to be NP-hard for weighted voters even when the number of candidates is bounded by a small constant [4]. For unweighted voters, nailing the complexity of coalitional manipulation proved to be more challenging. However, Faliszewski et al. [5] have recently established that this problem is hard for most variants of Copeland, and Zuckerman *et al* [13] showed that it is easy for Veto and Plurality with Runoff. Further, a very recent paper [12] makes substantial progress in this direction, showing, for example, that unweighted coalitional manipulation is hard for Maximin and Ranked Pairs, but easy for Bucklin (see Section 2 for the definitions of these rules).

All of these papers (as well as the classic work of Bartholdi et al. [1]) assume that the non-manipulators' preferences are publicly known. However, unlike the sincere voters, the manipulators are not endowed with preferences, i.e., orderings of candidates: they simply want to get a particular candidate elected. This model is somewhat unsatisfactory for two reasons. First, it departs from the standard model of manipulation considered by Gibbard [7] and Satterthwaite [10], in which the manipulator, too, has a preference over the candidates, and a manipulation is deemed successful if it leads to an election outcome that the manipulator prefers to the outcome of truthful voting. Second, it is asymmetric in its treatment of sincere voters and manipulators, and thus does not explain how the manipulating coalition forms. Indeed, it would be natural to expect that the manipulators start out by having the same type of preferences as sincere voters, and then some agents—those who are not satisfied with the current outcome and are willing to submit an insincere ballot—get together and decide to coordinate their efforts.

However, it is quite difficult to formalize this intuition so as to obtain a realistic model of how the manipulating coalition forms. In particular, it is not clear how the voters who are interested in manipulation should identify each other, and then reach an agreement which candidate to promote. Indeed, the latter decision seems to call for a voting procedure, and therefore is itself vulnerable to strategic behavior. Further, even assuming that suitable coalition formation and decision-making procedures exist, their practical implementation may be hindered by the absence of reliable two-way communication among the manipulators.

In a recent paper [11], Slinko and While put forward a model that provides a partial answer to these questions. They consider a setting where a single voter v announces his manipulative vote L (the truthful preferences of all agents are, as usual, common knowledge) to his set of associates F, i.e., the voters whose true preferences coincide with those of v. As a result, some of the voters in F switch to voting L, while others (as well as all voters not in F) vote truthfully. This can happen if, e.g., v's instructions are broadcast via an unreliable channel, i.e., some of the voters in F simply do not receive the announcement, or if some voters in F consider it unethical to vote non-truthfully. That is, in this model, the manipulating coalition always consists of voters with identical preferences (and thus the problem of which candidate to promote

Cite as: Title, Author(s), *Proc. of 9th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2010)*, van der Hoek, Kaminka, Luck and Sen (eds.), May, 10–14, 2010, Toronto, Canada, pp. XXX-XXX.

is trivially resolved), and, moreover, the manipulators always vote in the same way. However, v does not know how many voters will support him in his decision to vote L. Thus, he faces a dilemma: it might be the case that if x voters from F follow him, then the outcome improves, while if some $y \neq x$ voters from F switch to voting L, the outcome becomes even less desirable to v than the current alternative (we provide an example in Section 2). If v is conservatively-minded, in such situations he would choose not to manipulate at all. In other words, he would view L as a successful manipulation only if (1) there exists a subset $U \subseteq F$ such that if the voters in U switch to voting L, the outcome improves; (2) for any $W \subseteq F$, if the voters in W switch to voting L the outcome does not get worse. Paper [11] calls any manipulation that satisfies (1) and (2) *safe*. The main result of [11] is a generalization of the Gibbard-Satterthwaite theorem [7, 10] to safe manipulation: the authors prove that any onto, non-dictatorial voting rule with at least 3 alternatives is safely manipulable, i.e., there exists a profile in which at least one voter has a safe manipulation. However, paper [11] does not explore the computational complexity of the related problems.

In this paper, we focus on algorithmic complexity of safe manipulation. We first formalize the relevant computational questions and discuss some basic relationships between them. We then study the complexity of these questions for several classic voting rules, such as Plurality, k-approval, Bucklin, and Borda.

Now, these classic voting rules are defined as *voting correspondences*, i.e., they may produce more than one winner. In practice, however, whenever the goal is to select a single alternative, such ties need to be resolved. In the traditional model of manipulation, this does not usually cause major problems. Indeed, the standard approach is to ask whether the manipulator(s) can make the preferred candidate the unique winner or a co-winner, which is equivalent to considering a tie-breaking rule that is adversarial (respectively, favorable) to the manipulator(s). The algorithms/hardness results for the unique winner problem can usually be adapted for the co-winner problem and vice versa. (We note, however, that tie-breaking rules do matter in complex or multistage voting rules, such as Copeland [5] or STV [3]; also, the very first hardness-of-manipulation result [1] relies on a rather intricate tie-breaking rule).

In contrast, in our setting, properties of tie-breaking rules turn out to play an important role, since a manipulation must have a desirable effect across several profiles, i.e., for each possible number of manipulators. Intuitively, this means that a successful manipulation is easier to find if the tie-breaking rule behaves in a consistent manner. Thus, in our work we formalize what it means for a tie-breaking rule to be consistent (our class of "good" rules includes breaking the ties adversarially to the manipulator), and analyze the complexity of safe manipulation problems for both "good" and general tie-breaking rules, as well as for both weighted and unweighted voters.

Our analysis shows that the complexity of safe manipulation problems depends on both the voters' weights and the properties of the tie-breaking rules. For example, for consistent tie-breaking rules, we show that finding a safe manipulation is easy for k-approval and for Bucklin, even if the voters are weighted. In contrast, for Borda, finding a safe manipulation—or even checking that a given vote is safe—turns out to be hard for weighted voters even if the tie-breaking rule is very simple and the number of candidates is bounded by a small constant. We provide a table that summarizes our algorithmic results in the end of Section 8.

We then explore whether it is possible to extend the model of safe manipulation to settings where the manipulator may be joined by voters whose preferences differ from his own. Indeed, in real life a voter may follow advice to vote in a certain way if it comes from a person whose preferences are similar (rather than identical) to hers, or simply because she thinks that voting in this manner can be beneficial to her. For instance, in politics, a popular personality may influence many different voters at once by announcing his decision to vote in a particular manner. We propose two ways of formalizing this idea, which differ in their approach to defining the set of a voter's potential followers. However, we show that for both of our extended models the associated computational problems tend to be harder than in the model of [11].

We conclude the paper by summarizing our results and proposing several directions for future research.

2. PRELIMINARIES AND NOTATION

An *election* is given by a set of *candidates* (also referred to as *alternatives*) $C = \{c_1, \dots, c_m\}$ and a set of *voters* $V = \{1, \dots, n\}$. Each voter i is represented by his *preference* R_i, which is a total order over C; we will also refer to total orders over C as *votes*. For readability, we will sometimes denote the order R_i by $\succ_i$. The vector $\mathcal{R} = (R_1, \dots, R_n)$ is called a *preference profile*. We say that two voters i and j are of the same *type* if $R_i = R_j$; we write $V_i = \{j \mid R_j = R_i\}$.

A *voting rule* $\mathcal{F}$ is a mapping from the set of all preference profiles to the set of candidates; if $\mathcal{F}(\mathcal{R}) = c$, we say that c *wins* under $\mathcal{F}$ in $\mathcal{R}$. A voting rule is said to be *anonymous* if $\mathcal{F}(\mathcal{R}) = \mathcal{F}(\mathcal{R}')$, where $\mathcal{R}'$ is a preference profile obtained by permuting the entries of $\mathcal{R}$. To simplify the presentation, in this paper we consider anonymous voting rules only. In addition, we restrict ourselves to voting rules that are polynomial-time computable.

During the election, each voter i submits a vote L_i; the outcome of the election is then given by $\mathcal{F}((L_1, \dots, L_n))$. We say that a voter i is *truthful* if $L_i = R_i$. For any $U \subseteq V$ and a vote L, we denote by $\mathcal{R}_{-U}(L)$ the profile obtained from $\mathcal{R}$ by replacing R_i with L for all $i \in U$.

Voting rules We will now define the voting rules (or, more precisely, voting correspondences, see the discussion below) considered in this paper. All of these rules assign scores to all candidates; the winner(s) are the candidates with the highest scores.

Given a vector $\alpha = (\alpha_1, \dots, \alpha_k)$ with $\alpha_1 \geq \dots \geq \alpha_m$, the *score* $s_\alpha(c)$ of a candidate $c \in C$ under a *positional scoring rule* F_α is given by $\sum_{i \in V} \alpha_{j(i,c)}$, where $j(i, c)$ is the position in which voter i ranks candidate c. Many classic voting rules can be represented using this framework. Indeed, *Plurality* is the scoring rule with $\alpha = (1, 0, \dots, 0)$, *Veto* (also known as *Antiplurality*) is the scoring rule with $\alpha = (1, \dots, 1, 0)$, and *Borda* is the scoring rule with $\alpha = (m-1, m-2, \dots, 1, 0)$. Further, *$k$-approval* is the scoring rule with α given by $\alpha_1 = \dots = \alpha_k = 1$, $\alpha_{k+1} = \dots = \alpha_m = 0$; we will also refer to $(m-k)$-approval as *k-veto*.

Bucklin rule can be viewed as adaptive version of k-approval. We say that k, $1 \leq k \leq m$, is the *Bucklin winning round* if for any $j < k$ no candidate is ranked in top j positions by at least $\lceil n/2 \rceil$ voters, and there exists some candidate that is ranked in top k positions by at least $\lceil n/2 \rceil$ voters. We say that the candidate c's *score in round j* is his j-approval score, and his *Bucklin score* $s_B(c)$ is his k-approval score, where k is the Bucklin winning round. The *Bucklin winner* is the candidate with the highest Bucklin score. Observe that the Bucklin score of the Bucklin winner is at least $\lceil n/2 \rceil$.

Tie-breaking rules The classic voting rules defined above are, in fact, *voting correspondences*, as they may output several candidates as winners. A voting correspondence can be transformed into a voting rule by using a tie-breaking rule. Formally, a *tie-breaking rule* is a mapping T that given a preference profile $\mathcal{R}$ over C and a set $S \subseteq C$, selects a unique candidate $c \in S$; we write $T = T(\mathcal{R}, S)$. In what follows, to simplify notation, we will usually omit the first argument of T, and write $T = T(S)$; however, we emphasize that the tie-breaking rule may depend on voters' preferences. Denote by $f \circ g$ the *composition* of two functions f and g, i.e., $(f \circ g)(x) = f(g(x))$. For any voting correspondence $\mathcal{F}_0$, $\mathcal{F} = T \circ \mathcal{F}_0$ is a voting rule. Throughout this paper, we only consider tie-breaking rules that are polynomial-time computable.

We say that a tie-breaking rule T is *order-based* if there is an ordering $c_{i_1} \succ \ldots \succ c_{i_m}$ of the candidates in C such that given a set S, T always outputs an element of S that is maximal with respect to $\succ$. We say that an order-based rule T that corresponds to an ordering $\succ$ is *adversarial* to a voter i if $c_j \succ c_k$ if and only if $c_k \succ_i c_j$. It is easy to construct tie-breaking rules that are not order-based. For example, a tie-breaking rule T that satisfies $T(\{2,3\}) = 2$, $T(\{2,3,4\}) = 3$ is clearly not order-based. Some of our results only hold for order-based tie-breaking rules; in what follows, we will explicitly indicate if this is the case.

Weighted voters Our model can be extended to the situation where not all voters are equally important by assigning an integer *weight* w_i to each voter i. To compute the winner on a profile $(R_1, \ldots, R_n)$ under a voting rule $\mathcal{F}$ given voters' weights $\mathbf{w} = (w_1, \ldots, w_n)$, we apply $\mathcal{F}$ on a modified profile which for each $i = 1, \ldots, n$ contains w_i copies of R_i. As an input to our problems we usually get a *voting domain*, i.e., a tuple $S = \langle C, V, \mathbf{w}, \mathcal{R} \rangle$, together with a specific voting rule. When $\mathbf{w} = (1, \ldots, 1)$, we say that the voters are *unweighted*. For each $U \subseteq V$, let $|U|$ be the number of voters in U and let $w(U)$ be the total weight of the voters in U.

Safe manipulation We will now formally define the notion of safe manipulation. For the purposes of our presentation, we can simplify the definitions in [11] considerably.

As before, we assume that the voters' true preferences are given by a preference profile $\mathcal{R} = (R_1, \ldots, R_n)$.

DEFINITION 1. We say that a vote L is an *incentive to vote strategically*, or a *strategic vote* for i at $\mathcal{R}$ under $\mathcal{F}$, if $L \neq R_i$ and for some $U \subseteq V_i$ we have $\mathcal{F}(\mathcal{R}_{-U}(L)) \succ_i \mathcal{F}(\mathcal{R})$. Further, we say that L is a *safe strategic vote* for a voter i at $\mathcal{R}$ under $\mathcal{F}$ if L is a strategic vote at $\mathcal{R}$, and for any $U \subseteq V_i$ either $\mathcal{F}(\mathcal{R}_{-U}(L)) \succ_i \mathcal{F}(\mathcal{R})$ or $\mathcal{F}(\mathcal{R}_{-U}(L)) = \mathcal{F}(\mathcal{R})$.

To build intuition for the notions defined above, consider the following example.

EXAMPLE 1. Suppose $C = \{a, b, c, d\}$, $V = \{1, 2, 3, 4\}$, the first three voters have preference $b \succ a \succ c \succ d$, and the last voter has preference $c \succ d \succ a \succ b$. Suppose also that the voting rule is 2-approval combined with the lexicographic tie-breaking rule. Under truthful voting, a and b get 3 points, and c and d get 1 point each, so a wins. Now, if voter 1 changes his vote to $L = b \succ c \succ a \succ d$, b gets 3 points, a gets 2 points, and c gets 2 points, so b wins. As $b \succ_1 a$, L is a strategic vote for 1. However, it is not a safe strategic vote: if players in $V_1 = \{1, 2, 3\}$ all switch to voting L, then c gets 4 points, while b still gets 3 points, so in this case c wins and $a \succ_1 c$.

A *maximal* manipulation is one where all the voters from V_i choose to vote L. We will call the winner of such profile the *maximal manipulation winner* for L.

3. COMPUTATIONAL PROBLEMS: FIRST OBSERVATIONS

The definition of safe strategic voting gives rise to three natural algorithmic questions. In the definitions below, $\mathcal{F}$ is a given voting rule.

- ISSAFE($\mathcal{F}$): Given a voting domain, a voter i and a linear order L, is L a safe strategic vote for i under $\mathcal{F}$?
- EXISTSAFE($\mathcal{F}$): Given a voting domain and a voter i, can voter i make a safe strategic vote under $\mathcal{F}$?
- EXISTSAFEANYVOTER($\mathcal{F}$): Given a voting domain, can some voter make a safe strategic vote under $\mathcal{F}$?

Note that, in general, it is not clear if an efficient algorithm for EXISTSAFE($\mathcal{F}$) can be used to solve ISSAFE($\mathcal{F}$), or vice versa. However, if the number of candidates is constant, EXISTSAFE($\mathcal{F}$) reduces to ISSAFE($\mathcal{F}$).

PROPOSITION 1. *Consider any voting rule $\mathcal{F}$. For any constant k, if $|C| \leq k$, then a polynomial-time algorithm for* ISSAFE($\mathcal{F}$) *can be used to solve* EXISTSAFE($\mathcal{F}$) *is polynomial time.*

PROOF. In this case i has at most $k! = O(1)$ different votes, so he can try all of them. □

A similar reduction exists when each voter only has polynomially many "essentially different" votes.

PROPOSITION 2. *Consider any scoring rule $\mathcal{F}_\alpha$ that satisfies either (i) $\alpha_j = 0$ for all $j > k$ or (ii) $\alpha_j = 1$ for all $j \leq m - k$, where k is a given constant. For any tie-breaking rule T, a polynomial-time algorithm for* ISSAFE($T \circ \mathcal{F}_\alpha$) *can be used to solve* EXISTSAFE($T \circ \mathcal{F}_\alpha$) *is polynomial time.*

PROOF. We consider case (i); case (ii) is similar. There are at most $n^k = \text{poly}(n)$ different ways to fill the top k positions in a vote. Further, if two votes only differ in positions $k+1, \ldots, m$, they result in the same outcome. Thus, to solve EXISTSAFE($T \circ \mathcal{F}_\alpha$), it suffices to run ISSAFE($T \circ \mathcal{F}_\alpha$) on $\text{poly}(n)$ instances. □

Observe that the class of rules considered in Proposition 2 includes Plurality and Veto, as well as k-approval and k-veto when k is bounded by a constant.

Note also that EXISTSAFEANYVOTER($\mathcal{F}$) can be easily reduced to EXISTSAFE($\mathcal{F}$). Thus, in what follows, we focus on the first two problems in our list.

Unweighted voters When the voters are unweighted, i.e., $\mathbf{w} = (1, \ldots, 1)$, ISSAFE($\mathcal{F}$) is always easy.

PROPOSITION 3. *When voters are unweighted,* ISSAFE($\mathcal{F}$) *is in* P *for any voting rule $\mathcal{F}$.*

PROOF. Set $V_i = \{i_1, \ldots, i_s\}$. Since our voting rule is anonymous, it suffices to check the conditions of Definition 1 for $U \in \{\{i_1\}, \{i_1, i_2\}, \ldots, \{i_1, \ldots, i_s\}\}$, i.e., for $s \leq n$ sets of voters. □

Together with Propositions 1 and 2, Proposition 3 implies that the problem EXISTSAFE($\mathcal{F}$) is in P for any voting rule $\mathcal{F}$ that is obtained by combining Plurality, Veto, k-veto or k-approval for constant k with an arbitrary tie-breaking rule, as well as for any voting rule with a constant number of candidates.

Note that when voters are weighted, the conclusion of Proposition 3 no longer holds. Indeed, in this case the number of subsets of V_i that have different weights (and thus may have a different effect on the outcome) may be exponential in n.

4. PLURALITY, VETO, AND K-APPROVAL

We will now show that the easiness results for k-approval and k-veto extend to arbitrary $k \leq m$ and weighted voters as long as the underlying tie-breaking rule is order-based (note that the distinction between k-veto and $(m-k)$-approval only matters for constant k).

THEOREM 4. *The problems* ISSAFE($\mathcal{F}$) *and* EXISTSAFE($\mathcal{F}$) *are in* P *for any voting rule $\mathcal{F}$ obtained by composing some order-based tie-breaking rule T with k-approval.*

PROOF. Fix a voter $v \in V$ and assume without loss of generality that his preference order is given by $c_1 \succ \ldots \succ c_m$. Denote v's truthful vote by R. Recall that V_v denotes the set of voters who have the same preferences as v. Suppose that under truthful voting the winner is c_j. For $i = 1, \ldots, m$, let $s_i(\mathcal{R}')$ denote the k-approval score of c_i given a profile $\mathcal{R}'$, and set $s_i = s_i(\mathcal{R})$.

We start by proving a useful characterization of safe strategic votes for k-approval.

LEMMA 1. *A vote L is a safe strategic vote for v if and only if we have $\mathcal{F}(\mathcal{R}_{-V_v}(L)) = c_i$ for some $i < j$.*

PROOF. Suppose that L is a safe strategic vote for v. Then we have $\mathcal{F}(\mathcal{R}_{-U}(L)) = c_i$ for some $i < j$ and some $U \subseteq V_v$. It must be the case that each switch from R to L increases c_i's score or decreases c_j's score: otherwise c_i cannot beat c_j after the voters in U change their vote from R to L. Therefore, if c_i beats c_j when the preference profile is $\mathcal{R}_{-U}(L)$, it continues to beat c_j after the remaining voters in V_v switch, i.e., when the preference profile is $\mathcal{R}_{-V_v}(L)$. Hence, $\mathcal{R}_{-V_v}(L) \neq c_j$; since L is safe, this means that $\mathcal{R}_{-V_v}(L) = c_\ell$ for some $\ell < j$.

For the opposite direction, suppose that $\mathcal{F}(\mathcal{R}_{-V_v}(L)) = c_i$ for some $i < j$. Note that if two candidates gain points when some subset of voters switches from R to L, they both gain the same number of points; the same holds if both of them lose points.

Now, if $j > k$, a switch from R to L does not lower the score of c_j, so it must increase the score of c_i for it to be the maximal manipulation winner. Further, if a switch from R to L grants points to some $c_\ell \neq c_i$, then either $s_\ell < s_i$ or $s_\ell = s_i$ and the tie-breaking rule favors c_i over c_ℓ: otherwise, c_i would not be the maximal manipulation winner.

Similarly, if $j \leq k$, a switch from R to L does not increase the score of c_i, so it must lower the score of c_j. Further, if some $c_\ell \neq c_i$ does not lose points from a switch from R to L, then either $s_\ell < s_i$ or $s_\ell = s_i$ and the tie-breaking rule favors c_i over c_ℓ: otherwise, c_i would not be the maximal manipulation winner.

Now, consider any $U \subseteq V_v$. If

$$s_j(\mathcal{R}_{-U}(L)) > s_i(\mathcal{R}_{-U}(L)),$$

then by the argument above c_j is the winner. If

$$s_i(\mathcal{R}_{-U}(L)) > s_j(\mathcal{R}_{-U}(L)),$$

then by the argument above c_i is the winner. Finally, suppose

$$s_i(\mathcal{R}_{-U}(L)) = s_j(\mathcal{R}_{-U}(L)).$$

By the argument above, no other candidate can have a higher score. So, suppose that $s_\ell(\mathcal{R}_{-U}(L)) = s_i(\mathcal{R}_{-U}(L))$, and the tie-breaking rule favors c_ℓ over c_i and c_j. Then this would imply that c_ℓ wins in $\mathcal{R}$ or $\mathcal{R}_{-V_v}(L)$ (depending on whether a switch from R to L causes c_ℓ to lose points), a contradiction. Thus, in this case, too, either c_i or c_j wins. □

Lemma 1 immediately implies an algorithm for ISSAFE($\mathcal{F}$): we simply need to check that the input vote satisfies the conditions of the lemma. We now show how to use it to construct an algorithm for EXISTSAFE($\mathcal{F}$). We need to consider two cases.

j > k:
In this case, the voters in V_v already do not approve of c_j and approve of all c_i, $i \leq k$. Thus, no matter how they vote, they cannot ensure that some c_i, $i \leq k$, gets more points than c_j. Hence, the only way they can change the outcome is by approving of some candidate c_i, $k < i < j$. Further, they can only succeed if there exists an $i = k+1, \ldots, j-1$ such that either $s_i + w(V_v) > s_j$ or $s_i + w(V_v) = s_j$ and the tie-breaking rule favors c_i over c_j. If such an i exists, v has an incentive to manipulate by swapping c_1 and c_i in his vote. Furthermore, it is easy to see that any such manipulation is safe, as it only affects the scores of c_1 and c_i.

j ≤ k:
In this case, the voters in V_v already approve of all candidates they prefer to c_j, and therefore they cannot increase the scores of the first $j-1$ candidates. Thus, their only option is to try to lower the scores of c_j as well as those of all other candidates whose score currently matches or exceeds the best score among $s_1, \ldots, s_{j-1}$.

Set $C_g = \{c_1, \ldots, c_{j-1}\}$, $C_b = \{c_j, \ldots, c_m\}$. Let C_0 be the set of all candidates in C_g whose k-approval score is maximal, and let $s_{\max}$ be the k-approval score of the candidates in C_0. For any $c_\ell \in C_b$, let s'_ℓ denote the number of points that c_ℓ gets from all voters in $V \setminus V_v$; we have $s'_\ell = s_\ell$ for $k < \ell \leq m$ and $s'_\ell = s_\ell - w(V_v)$ for $\ell = j, \ldots, k$. Now, it is easy to see that v has a safe manipulation if and only if the following conditions hold:

- For all $c_\ell \in C_b$ either $s'_\ell < s_{\max}$, or $s'_\ell = s_{\max}$ and there exists a candidate $c \in C_0$ such that the tie-breaking rule favors c over c_ℓ;
- There exist a set $C_{\text{safe}} \subseteq C_b$, $|C_{\text{safe}}| = k - j + 1$, such that for all $c_\ell \in C_{\text{safe}}$ either $s'_\ell + w(V_v) < s_{\max}$ or $s'_\ell + w(V_v) = s_{\max}$ and there exists a candidate $c \in C_0$ such that the tie-breaking rule favors c over c_ℓ.

Note that these conditions can be easily checked in polynomial time by computing s_ℓ and s'_ℓ for all $\ell = 1, \ldots, m$.

Indeed, if such a set C_{safe} exists, voter v can place the candidates in C_{safe} in positions $j, \ldots, k$ in his vote; denote the resulting vote by L. Clearly, if all voters in V_v vote according to L, they succeed to elect some $c \in C_0$. Thus, by Lemma 1, L is safe. Conversely, if a set C_{safe} with these properties does not exist, then for any vote $L \neq R$ we have $\mathcal{F}(\mathcal{R}_{-V_v}(L)) \in C_b$, and thus by Lemma 1 L is not safe. □

Theorem 4 hold for order-based tie-breaking rules. In contrast, for arbitrary tie-breaking rules finding a safe vote with respect to k-approval is NP-hard (assuming k is viewed as a part of the input).

THEOREM 5. *There is an efficiently computable tie-breaking rule T such that* EXISTSAFE *is* NP-*hard for the composition of T and k-approval. This result holds even if voters are unweighted.*

The proof is omitted due to space constraints.

Theorem 5 relies on having a large number of candidates, and only proves the hardness of EXISTSAFE($\mathcal{F}$) (indeed, as argued above, for unweighted voters ISSAFE($\mathcal{F}$) is always easy). In contrast, if we allow weighted voters in addition to arbitrary tie-breaking rules, then both ISSAFE($\mathcal{F}$) and EXISTSAFE($\mathcal{F}$) are hard even when the number of candidates is bounded by a constant (which may, however, depend on the actual voting rule, i.e., on k).

THEOREM 6. *There exists a (non-order-based) tie-breaking rule T such that* ISSAFE($\mathcal{F}$) *is* coNP-*hard for the composition of T and k-approval for any $k \geq 2$ and any $m \geq k+3$.*

The proof is omitted due to space constraints. Later, we will show how to generalize this result to EXISTSAFE (Corollary 12).

REMARK 1. Note that we do not claim that ISSAFE is in coNP for k-approval with weighted voters: indeed, to prove that L is not a safe strategic vote for i, we need to either guess *some* coalition $U \subseteq V_i$ such that $\mathcal{F}(\mathcal{R}) \succ_i \mathcal{F}(\mathcal{R}_{-U}(L))$, or to prove that for *any* $U \subseteq V_i$ we have $\mathcal{F}(\mathcal{R}) = \mathcal{F}(\mathcal{R}_{-U}(L))$. Thus, this problem is in Δ_p^2, but it is not clear if it is in coNP (we refer the reader to e.g., [9] for the definition of the class Δ_p^2). The same holds true for other hardness results discussed later in the paper.

While the proof of Theorem 6 does not go through for Plurality or Veto, the argument can be adapted to work for both of these voting rules. In particular, for Plurality this implies that IsSAFE is coNP-hard even for 4 candidates. We will now show that this result is tight: any scoring rule with 3 candidates is easy to manipulate safely, even if the voters are weighted and arbitrary tie-breaking rules are allowed.

THEOREM 7. *IsSAFE($\mathcal{F}$) is in* P *for any voting rule $\mathcal{F}$ obtained by combining a positional scoring rule with at most three candidates with an arbitrary tie-breaking rule.*

The proof proceeds by case analysis and is omitted.

5. BUCKLIN RULE

Bucklin rule is quite similar to k-approval, so we can use the ideas in the proof of Theorem 4 to obtain a similar result for Bucklin. However, the proof becomes significantly more complicated.

THEOREM 8. *EXISTSAFE($\mathcal{F}$) is in* P *for any voting rule $\mathcal{F}$ obtained by composing an arbitrary order-based tie-breaking rule T with Bucklin.*

PROOF. Fix a voter $v \in V$ and assume without loss of generality that his preference order is given by $c_1 \succ \ldots \succ c_m$. Denote v's truthful vote by R. Let V_v denote the set of voters who have the same preferences as v. Suppose that under truthful voting the Bucklin winner is c_j, and the winning round is k.

We have to consider two possibilities.

$\mathbf{j > k}$:
In this case, no matter how the voters in V_v vote, c_j's k-approval score will be at least $\lceil n/2 \rceil$. Thus, the only part of v's vote that can affect the final outcome is the top k positions. Now, no matter how the voters in V_v vote, none of the candidates currently ranked in the top k positions by v can beat c_j. Thus, the only way v can succeed is by ranking some candidate c_i, $k < i < j$, in the top position. This will work if it makes c_i's k'-approval score become at least $\lceil n/2 \rceil$ for some $k' < k$, or if it makes c_i a k-approval winner (under the given tie-breaking rule). To check if such a c_i exists, v can try to swap c_1 and c_i in his vote for all $i = k+1, \ldots, j-1$. Moreover, any such manipulation is safe, as it only affects the scores of c_1 and c_i.

$\mathbf{j \le k}$:
Set $C_g = \{c_1, \ldots, c_{j-1}\}$, $C_b = \{c_j, \ldots, c_m\}$. Let k' be the smallest value of ℓ such that under truthful voting some $c \in C_g$ gets at least $\lceil n/2 \rceil$ votes in round ℓ; clearly, we have $k' \ge k$. Let C_0 be the set of all candidates in C_g whose k'-approval score is maximal (and hence is at least $\lceil n/2 \rceil$), and let $s_{\max}$ be the k'-approval score of the candidates in C_0. Let $\widehat{c}$ be the candidate in C_0 that is most favored by the tie-breaking rule. Throughout the proof, for any $c, c' \in C$ and any $\ell \le m$, we will say that c beats c' under ℓ-approval if c's ℓ-approval score is no lower than that of c', and if they are equal, then our tie-breaking rule favors c over c'. Further, we say that c beats c' if there exist some $\ell, \ell' \le m$, such that c gets at least $\lceil n/2 \rceil$ votes under ℓ-approval, but not under $(\ell-1)$-approval, c' gets at least $\lceil n/2 \rceil$ votes under ℓ'-approval, but not under $(\ell'-1)$-approval, and either $\ell < \ell'$ or $\ell = \ell'$ and c beats c' under ℓ-approval.

No matter how the voters in V_v vote, they cannot ensure that a candidate in C_g gets at least $\lceil n/2 \rceil$ votes in round ℓ for $\ell < k'$. Therefore, to succeed, they need to ensure that no candidate in C_b gets at least $\lceil n/2 \rceil$ ℓ-approval votes for $\ell < k'$, and that no candidate in C_b beats all candidates in C_g under k'-approval.

Hence, v has an incentive to manipulate if and only if there is a vote L such that if some voters in V_v switch from R to L, then for any $c \in C_b$

(a) c's $(k'-1)$-approval score is less than $\lceil n/2 \rceil$;

(b) c's k'-approval is at most $s_{\max}$, and if it is equal to $s_{\max}$, then the tie-breaking rule favors $\widehat{c}$ over c.

We will first claim that essentially any safe vote L makes c_j lose when all voters in V_v switch from R to L.

LEMMA 2. *If there exists a safe vote for v, then there exists a safe vote for v that ranks all candidates in C_g in top $j-1$ positions. Moreover, for any safe vote L that ranks all candidates in C_g in top $j-1$ positions, it cannot be the case that c_j wins if all voters in V_v vote according to L.*

We omit the proof. We will now construct a family of votes for v as follows. For $\ell = j, \ldots, m$, let us say that a position $p \in \{j, \ldots, m\}$ is *safe* for c_ℓ if c_ℓ satisfies conditions (a) and (b) above whenever all voters in V_v rank c_ℓ in position p. Let $P(\ell)$ denote the set of positions that are safe for c_ℓ. The sets $P(\ell)$ can be computed independently and efficiently for each $c_\ell \in C_b$. Consider now a bipartite graph G whose vertices are candidates in C_b and positions in $\{j, \ldots, m\}$, and there is an edge from a candidate c_ℓ to a position i if and only if $i \in P(\ell)$. For any complete matching in this graph, we can construct a vote L in which all candidates in C_g are ranked in top $j-1$ positions and all candidates in C_b are ranked according to the matching. Denote the set of all such votes by $\mathcal{L}(G)$. Clearly, if all voters in V_v vote according to any $L \in \mathcal{L}(G)$, then some candidate in C_g wins. We will now prove two lemmas that characterize the relationship between the set $\mathcal{L}(G)$ and the set of all safe votes.

LEMMA 3. *Any safe vote L that ranks all candidates in C_g in top $j-1$ positions ranks each $c_\ell \in C_b$ in a position in $P(\ell)$.*

PROOF. Suppose that L ranks some $\ell \in C_b$ in a position that is not in $P(\ell)$. This means that no candidate in C_g can win when all voters in V_v vote according to L. Now, if in this situation some $c' \in C_b$, $c' \neq c_j$, wins, this means that L is unsafe. On the other hand, if c_j wins, then by Lemma 2 L is not safe. □

Thus, each safe vote can be transformed into a vote in $\mathcal{L}(G)$, i.e., if $\mathcal{L}(G) = \emptyset$, there are no safe votes for v.

LEMMA 4. *If there exists a safe vote for v, then any vote in $\mathcal{L}(G)$ is safe.*

The proof proceeds by contraposition: we will argue that if some vote $L \in \mathcal{L}(G)$ is not safe, then no vote is safe for v. We omit the proof due to space constrains.

Thus, we conclude that to check whether v has a safe vote (and to find one if it exists), it suffices to compute the set $\mathcal{L}(G)$, check if it is not empty, and, if it contains some $L \in \mathcal{L}(G)$, check if L is safe. Indeed, we have argued that if $\mathcal{L}(G) = \emptyset$ then no vote is safe, and if $\mathcal{L}(G) \neq \emptyset$ then v has a safe vote if and only if an arbitrary vote in $\mathcal{L}(G)$ is safe. □

Unlike in the case of k-approval, it is not clear if the same approach can be used to show that IsSAFE for Bucklin with order-based tie-breaking is in P for weighted voters.

On the other hand, if we allow arbitrary tie-breaking rules, then IsSAFE becomes hard for Bucklin even with 5 candidates. The proof of the following theorem is omitted.

THEOREM 9. *There exists a (non-order-based) tie-breaking rule T such that* IsSAFE($T \circ$ Bucklin) *is* coNP-*hard. This holds even if there are only* 5 *candidates.*

6. BORDA RULE

Unlike k-approval and Bucklin, Borda is hard to manipulate safely even if we restrict ourselves to order-based tie-breaking rules. Due to space constraints, all proofs in this section are omitted.

THEOREM 10. *There exists an order-based tie-breaking rule T such that* $\text{IsSAFE}(T \circ \text{Borda})$ *is* coNP*-hard. The hardness result holds even if there are only* 5 *candidates.*

If we allow arbitrary tie-breaking rules, then ISSAFE becomes hard for Borda even with 4 candidates.

THEOREM 11. *There exists a (non-order-based) tie-breaking rule T such that* $\text{IsSAFE}(T \circ \text{Borda})$ *is* coNP*-hard. This holds even if there are only* 4 *candidates.*

Observe that by Theorem 7 this result is tight with respect to the number of candidates.

Finally, we note that, in general, hardness results for ISSAFE do not imply similar hardness results for EXISTSAFE. However, if the strategic vote constructed in the hardness proof for ISSAFE is the only strategic vote available to the manipulator, then the same construction can be used to show the hardness of EXISTSAFE. It can be verified that this is the case for all hardness proofs in this paper Thus, we obtain the following corollary.

COROLLARY 12. $\text{EXISTSAFE}(\mathcal{F})$ *is* coNP*-hard if the voting rule $\mathcal{F}$ satisfies the conditions of Theorems 6, 9, 10, or 11.*

7. EXTENSIONS OF THE SAFE STRATEGIC VOTING MODEL

Throughout the paper, we followed the model of [11] and assumed that the only voters who may change their votes are the ones whose preferences exactly coincide with those of the manipulator. Clearly, in real life this assumption does not always hold. Indeed, a voter may follow a suggestion to vote in a certain way as long as it comes from someone he trusts (e.g., a well-respected public figure), and this does not require that this person's preferences are completely identical to those of the voter. For example, if both the original manipulator i and his would-be follower j rank the current winner last, it is easy to see that following i's recommendation that leads to displacing the current winner is in j's best interests.

In this section, we will consider two approaches to extending the notion of safe strategic voting to scenarios where not all manipulators have identical preferences. In both cases, we define the set of potential followers for each voter (in our second model, this set may depend on the vote suggested), and define a vote L to be safe if, whenever a subset of potential followers votes L, the outcome of the election does not get worse (and sometimes get better) from the manipulator's perspective. However, our two models differ in the criteria they use to identify a voter's potential followers. Due to space constraints, all proofs in this section are omitted.

Preference-Based Extension Our first model identifies the followers of a given voter based on the similarities in voters' preferences.

Fix a preference profile $\mathcal{R}$ and a voting rule $\mathcal{F}$, and let c_w be the winner under truthful voting. For any $i = 1, \ldots, n$, let $I(i, w)$ denote the set of candidates that i ranks strictly above c_w. We say that two voters i and j are *similar* if $I(i, w) = I(j, w)$. A *similar set* S_i of a voter i for a given preference profile $\mathcal{R}$ and a voting rule $\mathcal{F}$ is given by $S_i = \{j \mid I(i, w) = I(j, w)\}$. (The set S_i depends on $\mathcal{R}$ and $\mathcal{F}$; however, for readability we omit $\mathcal{R}$ and $\mathcal{F}$ from the notation).

Note that if i and j are similar, they rank c_w in the same position. Further, a change of outcome from c_w to another alternative is positive from i's perspective if and only if it is positive from j's perspective. Thus, intuitively, any manipulation that is profitable for i is also profitable for j. Observe also that similarity is an equivalence relation, and the sets S_i are the corresponding equivalence classes. In particular, this implies that for any $i, j \in n$ either $S_i = S_j$ or $S_i \cap S_j = \emptyset$.

We can now adapt Definition 1 to our setting by replacing V_i with S_i.

DEFINITION 2. *A vote L is a* strategic vote in the preference-based extension *for i at $\mathcal{R}$ under $\mathcal{F}$ if $L \neq R_i$ and for some $U \subseteq S_i$ we have $\mathcal{F}(\mathcal{R}_{-U}(L)) \succ_i \mathcal{F}(\mathcal{R})$. Further, we say that L is a* safe strategic vote in the preference-based extension *for a voter i at $\mathcal{R}$ under $\mathcal{F}$ if L is a strategic vote at $\mathcal{R}$ under $\mathcal{F}$, and for any $U \subseteq S_i$ either $\mathcal{F}(\mathcal{R}_{-U}(L)) \succ_i \mathcal{F}(\mathcal{R})$ or $\mathcal{F}(\mathcal{R}_{-U}(L)) = \mathcal{F}(\mathcal{R})$.*

Observe that if L is a (safe) strategic vote for i at $\mathcal{R}$ under $\mathcal{F}$, then it is also a (safe) strategic vote for any $j \in S_i$. Indeed, $j \in S_i$ implies $S_i = S_j$ and for any $c \in C$ we have $c \succ_i \mathcal{F}(\mathcal{R})$ if and only if $c \succ_j \mathcal{F}(\mathcal{R})$.

Clearly, for any voter i, the set S_i of similar voters is easy to compute. However, the problems that we considered throughout this paper, i.e., the safety of a given manipulation and the existence of a safe manipulation become considerably more difficult in the preference-based extension. Indeed, while voters in S_i have similar preferences, their truthful votes may be substantially different, so it now matters *which* of the voters in S_i decide to follow the manipulator (rather than just *how many* of them, as in the original model). To illustrate this, we will now demonstrate that in the preference-based extension deciding if a particular vote is safe under 3-approval is computationally difficult. In contrast, we have seen that in the original model this question is easy for k-approval with any constant k.

THEOREM 13. *There exists a polynomial-time tie-breaking rule T such that for $T \circ 3$-approval it is* coNP*-hard to decide whether a given vote is a a safe strategic vote in the preference-based extension.*

In the preference-based model, a voter i will follow a recommendation to vote in a particular way if it comes from a voter whose preferences are similar to those of i. However, this approach does not describe settings where a voter follows a recommendation not so much because he trusts the recommender, but for pragmatic purposes, i.e., because the proposed manipulation advances her own goals. Clearly, this may happen even if the overall preferences of the original manipulator and the follower are substantially different. We will now propose a model that aims to capture this type of scenarios.

Goal-Based Extension If the potential follower's preferences are different from those of the manipulator, his decision to join the manipulating coalition is likely to depend on the specific manipulation that is being proposed. Thus, in this subsection we will define the set of potential followers F in a way that depends both on the original manipulator's identity i and his proposed vote L, i.e., we have $F = F_i(L)$. Note, however, that it is not immediately obvious how to decide whether a voter j can benefit from following i's suggestion to vote L, and thus should be included in the set $F_i(L)$. Indeed, the benefit to j depends on which other voters are in the set $F_i(L)$, which indicates that the definition of the set $F_i(L)$ has to be self-referential.

In more detail, for a given voting rule $\mathcal{F}$, an election (C, V) with a preference profile $\mathcal{R}$, a voter $i \in V$ and a vote L, we say that a voter j is *pivotal for a set $U \subseteq V$ with respect to* (i, L) if $j \notin U$, $R_j \neq L$ and $\mathcal{F}(\mathcal{R}_{-(U \cup \{j\})}(L)) \succ_j \mathcal{F}(\mathcal{R}_{-U}(L))$. That is, a voter j is pivotal for a set U if when the voters in U vote according to

L, it is profitable for j to join them. Now, it might appear natural to define the follower set for (i, L) as the set that consists of i and all voters $j \in V$ that are pivotal with respect to (i, L) for some set $U \subseteq V$. However, this definition is too broad: a voter is included as long as it is pivotal for some subset $U \subseteq V$, even if the voters in U cannot possibly benefit from voting L. To exclude such scenarios, we need to require that U itself is also drawn from the follower set. Formally, we say that $F_i(L)$ is a *follower set* for (i, L) if it is a maximal set F that satisfies the following condition:

$$\forall j \in F \, [(j = i) \vee (\exists U \subseteq F \text{ s. t. } j \text{ is pivotal for } U \text{ wrt } (i, L))] \; (*)$$

Note that this means that $F_i(L)$ is a fixed point of a mapping from 2^V to 2^V, i.e., this definition is indeed self-referential. To see that the follower set is uniquely defined for any $i \in V$ and any vote L, note that the union of any two sets that satisfy condition $(*)$ also satisfies $(*)$.

We can now define what it means for L to be a *strategic vote in the goal-based extension* and a *safe strategic vote in the goal-based extension* by replacing the condition $U \subseteq S_i$ with $U \subseteq F_i(L)$ in Definition 2. However, this definition of a safe vote is not completely satisfying, as it only guarantees safety to the original manipulator, but not to her followers. (Note that this issue did not arise in the preference-based extension, since any vote that was safe for the original manipulator was also safe for all similar voters.) If we expect the followers to be as cautious as the original manipulator, we need to modify the previous definition by additionally requiring that for all $j \in F_i(L)$ it holds that $\mathcal{F}(\mathcal{R}_{-U}(L)) \succ_j \mathcal{F}(\mathcal{R})$ or $\mathcal{F}(\mathcal{R}_{-U}(L)) = \mathcal{F}(\mathcal{R})$ for any $U \subseteq F_i(L)$; if this holds, we say that L is a *universally safe strategic vote for* (i, L) *in the goal-based extension*.

The definition of a (universally) safe strategic vote in the goal-based extension captures a number of situations not accounted for by the definition of a safe strategic vote in the preference-based extension. To see this, consider the following example.

EXAMPLE 2. Consider an election with the set of candidates $C = \{a, b, c, d, e\}$, and three voters 1, 2, and 3, whose preferences are given by $a \succ_1 b \succ_1 c \succ_1 d \succ_1 e$, $e \succ_2 b \succ_2 a \succ_2 d \succ_2 c$, and $d \succ_3 a \succ_3 b \succ_3 c \succ_3 e$. Suppose that the voting rule is Plurality combined with an order-based tie-breaking rule T with the underlying order $d \succ b \succ c \succ e \succ a$.

Under truthful voting, d is the winner, so we have $S_1 \neq S_2$. Thus, in the preference-based extension, a vote that ranks a first is a safe strategic vote for voter 2, but a vote than ranks b first is not. On the other hand, let L be any vote that ranks b first. Then $F_1(L) = F_2(L) = \{1, 2\}$. Indeed, if voter 1 switches to voting L, the winner is still d, but it becomes profitable for voter 2 to join her, and vice versa. On the other hand, it is easy to see that voter 3 cannot profit by voting L. It follows that in the goal-based extension L is a universally safe strategic vote for voter 1.

From a practical perspective, it is plausible that in Example 2 voters 1 and 2 would be able to reconcile their differences (even though they are substantial—voter 1 ranks voter 2's favorite candidate last) and jointly vote for b, as this provides a significant improvement for both of them. Thus, at least in some situations the model provided by the goal-based extension is intuitively more appealing. However, computationally it is even more intractable than the preference-based extension.

Indeed, it is not immediately clear how to compute the set $F_i(L)$, as its definition is non-algorithmic in nature. While one can consider all subsets of V and check whether they satisfy condition $(*)$, this approach is obviously inefficient. We can avoid full enumeration if have access to a procedure $\mathcal{A}(i, L, j, W)$ that for each pair (i, L), each voter $j \in V$ and each set $W \subseteq V$ can check if $j = i$ or there is a set $U \subseteq W$ such that j is pivotal for U with respect to (i, L). Indeed, if this is the case, we can compute $F_i(L)$ as follows. We start with $W = V$, run $\mathcal{A}(i, L, j, W)$ for all $j \in W$, and let W' to be the set of all voters for which $\mathcal{A}(i, L, j, W)$ outputs "yes". We then set $W = W'$, and iterate this step until $W = W'$. In the end, we set $F_i(L) = W$. The correctness of this procedure can be proven by induction on the number of iterations and follows from the fact that if a set W contains no subset U that is pivotal for j, then no smaller set $W' \subset W$ can contain such a subset. Moreover, since each iteration reduces the size of W, the process converges after at most n iterations. Thus, this algorithm runs in polynomial time if the procedure $\mathcal{A}(i, L, j, W)$ is efficiently implementable. We will now show that this is indeed the case for Plurality combined with an order-based tie-breaking rule.

THEOREM 14. *Given an order-based tie-breaking rule T, an election (C, V) with a preference profile $\mathcal{R}$, a manipulator i, a vote L, and a voter j, we can compute the set $F_i(L)$ with respect to the voting rule $T \circ$Plurality in time polynomial in the input size.*

However, if we combine Plurality with an arbitrary tie-breaking rule, computing the set $F_i(L)$ becomes difficult.

THEOREM 15. *There exists a polynomial-time computable tie-breaking rule T such that for $T \circ$Plurality, given an election (C, V), a manipulator i, a vote L, and a voter j, it is* NP*-hard to decide whether $j \in F_i(L)$.*

Moreover, in the goal-based extension, finding a safe manipulation for 3-approval is hard even if we restrict ourselves to order-based tie-breaking rules; recall that in the standard model safely manipulating k-approval combined with an order-based tie-breaking rule is easy for arbitrary k.

THEOREM 16. *There is a poly-time computable order-based tie-breaking rule T such that for $T \circ$ 3-approval it is* coNP*-hard to decide whether a given vote is a safe strategic vote in the goal-based extension.*

To summarize, the results of this section show that there can be several approaches to extending the notion of safe strategic voting to non-identical manipulators. However. for both of the extended models that we considered, natural algorithmic questions become hard even for very simple voting rules, such as 3-approval or Plurality. Thus, it appears that from algorithmic perspective the original model of [11] is more interesting to study than its more expressive generalizations.

8. CONCLUSIONS

In this paper, we started the investigation of algorithmic complexity of safe manipulation, as defined by Slinko and While [11]. Our results for the model of [11] are summarized in Table 1. They indicate a rather complicated relationship between the complexity of our problems, the voters' weights and the type of the tie-breaking rule allowed. In particular, it appears that for the complexity of ISSAFE the most important parameter is whether the voters are weighted. In contrast, for the complexity of EXISTSAFE, the type of the tie-breaking rule seems to be more significant than the weights; however, for some voting rules allowing non-unit weights makes this problem hard even for a small number of candidates.

The reader may have noticed that our easiness proofs for order-based tie-breaking rules make use of an *independence of irrelevant alternatives property (IIA property)*: our proofs go through as long as the tie-breaking rule T is such that if $T(S) = c$ then $T(S') \in \{S' \setminus S\} \cup \{c\}$ for any $S' \supset S$. While this property

Table 1: Summary of results—identical manipulators

<table>
<tr><td rowspan="3">Voters</td><td rowspan="3"># of candidates</td><td rowspan="3">Rule</td><td colspan="4">Problem</td></tr>
<tr><td colspan="2">ISSAFE</td><td colspan="2">EXISTSAFE</td></tr>
<tr><td>order-based</td><td>any</td><td>order-based</td><td>any</td></tr>
<tr><td rowspan="5">weighted</td><td rowspan="5">constant or any</td><td>Plurality</td><td>P</td><td>coNP-hard</td><td>P</td><td>coNP-hard</td></tr>
<tr><td>Veto</td><td>P</td><td>coNP-hard</td><td>P</td><td>coNP-hard</td></tr>
<tr><td>k-approval</td><td>P</td><td>coNP-hard</td><td>P</td><td>coNP-hard</td></tr>
<tr><td>Bucklin</td><td></td><td>coNP-hard</td><td>P</td><td>coNP-hard</td></tr>
<tr><td>Borda</td><td colspan="4">coNP-hard</td></tr>
<tr><td rowspan="10">unweighted</td><td rowspan="5">constant</td><td>Plurality</td><td colspan="4" rowspan="5">P</td></tr>
<tr><td>Veto</td></tr>
<tr><td>k-approval</td></tr>
<tr><td>Bucklin</td></tr>
<tr><td>Borda</td></tr>
<tr><td rowspan="5">any</td><td>Plurality</td><td colspan="2" rowspan="5">P</td><td colspan="2">P</td></tr>
<tr><td>Veto</td><td colspan="2">P</td></tr>
<tr><td>k-approval</td><td>P</td><td>NP-hard</td></tr>
<tr><td>Bucklin</td><td>P</td><td></td></tr>
<tr><td>Borda</td><td></td><td></td></tr>
</table>

seems more general than the existence of an underlying order, the two conditions are, in fact, equivalent: a tie-breaking rule has the IIA property if and only if it is order-based (see [8]). We chose to present our results in terms of order-based rules since an order gives a poly-size representation for the tie-breaking rule; the importance of compact representation is illustrated by Theorem 5.

Observe also that while non-order-based rules may appear unnatural, they can arise if the ties are resolved by another voting rule (this method of tie-breaking is quite well-known, see, e.g., [1]). Another situation where ties may be broken in a non-IIA manner is when the alternatives belong to certain groups, and we are required to choose an alternative from a given group whenever the majority of the tied alternatives belongs to this group. This situation may arise in politics, where it can be used to promote certain underrepresented minorities, as well as in plan selection scenarios, where the alternative space may be structured, and we want to select a plan with a certain feature if the majority of the most popular plans have this feature.

We also proposed two ways of extending the notion of safe manipulation to heterogeneous groups of manipulators, and initiated the study of computational complexity of related questions. We demonstrated that taking into account the possibility that the group of manipulators may include voters whose preferences differ from those of the leader results in a substantial increase in computational complexity of safe manipulation problems. Due to space limitations, in this part of the paper, we focused on two simple voting rules and unweighted voters. In the future, it would be interesting to see which of our results can be extended to other voting rules and weighted voters.

Another open question is determining the complexity of finding a safe strategic vote for voting rules not considered in this paper, such as Copeland, Ranked Pairs, or Maximin. Moreover, for some of the voting rules we have investigated, the picture given by this paper is incomplete. In particular, it would be interesting to understand the computational complexity of finding a safe manipulation for Borda (and, more generally, for all scoring rules) in the simplest possible scenario, i.e., for unweighted voters and order-based tie-breaking. The problem for Borda is particularly intriguing as this is perhaps the only widely studied voting rule for which the complexity of unweighted coalitional manipulation in the standard model is not known.

Other exciting research directions include formalizing and investigating the problem of selecting the best safe manipulation (is it the one that succeeds more often, or one that achieves better results when it succeeds?), and extending our analysis to randomized tie-breaking rules. However, the latter question may require modifying the notion of a safe manipulation, as the outcome of a strategic vote becomes a probability distribution over the alternatives.

9. REFERENCES

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

[2] J. J. Bartholdi, III and J. B. Orlin. Single transferable vote resists strategic voting. *Social Choice and Welfare*, 8(4):341–354, 1991.

[3] V. Conitzer, M. Rognlie, L. Xia. Preference Functions that Score Rankings and Maximum Likelihood Estimation, in *Proc. IJCAI-09*, pp. 109–115, 2009.

[4] V. Conitzer, T. Sandholm, and J. Lang, When ere elections with few candidates hard to manipulate?, *J. ACM*, 54:1–33.

[5] P. Faliszewski, E. Hemaspaandra, and H. Schnoor. Copeland voting: ties matter, in *Proc. of AAMAS-08*, 2008.

[6] M. R. Garey, D. S. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness, 1979.

[7] A. F. Gibbard. Manipulation of voting schemes: a general result, *Econometrica*, 41:597–601, 1973.

[8] H. Moulin. Choice functions over a finite set: a summary, *Social Choice and Welfare*, 2:147–160, 1985.

[9] C. Papadimitriou. Computational Complexity, 1994.

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

[11] A. Slinko, S. White. Is it ever safe to vote strategically?, in *Proc. of COMSOC-08*, 2008.

[12] L. Xia, M. Zuckerman, A. D. Procaccia, V. Conitzer, and J. S. Rosenschein. Complexity of unweighted coalitional manipulation under some common voting rules, in *Proc. of IJCAI-09*, 2009.

[13] M. Zuckerman, A. D. Procaccia, and J. S. Rosenschein. Algorithms for the coalitional manipulation problem, *Artificial Intelligence*, 173(2):392–412, 2009.