Complexity of Safe Strategic Voting[1]

Noam Hazon and Edith Elkind

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, providing algorithms and hardness results for both weighted and unweighted voters. 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.

1 Introduction

Computational aspects of voting, and, in particular, voting manipulation, is an active topic of current research. 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 set of manipulators is given exogenously, and the manipulators are not endowed with preferences over the entire set of candidates; rather, they simply want to get a particular candidate elected, and select their votes based on the non-manipulators' preferences that are publicly known. That is, this model abstracts away the question of how the manipulating coalition forms. However, to develop a better understanding of coalitional manipulation, it is desirable to have a plausible model of the coalition formation process. In such a model the manipulators would 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—would 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

[1] To appear in the proceedings of the 3rd International Symposium on Algorithmic Game Theory (SAGT'10)

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 White 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. Such a situation is not unusual in politics, where a public figure may announce her decision to vote in a particular manner, and may be followed by a subset of like-minded people. That is, in this model, the manipulating coalition always consists of voters with identical preferences (and thus the problem of which candidate to promote is trivially resolved), and, moreover, the manipulators always vote in the same way. Further, it relies on minimal communication, i.e., a single broadcast message. However, due to lack of two-way communication, 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 the first part of this paper, we focus on algorithmic complexity of safe manipulation, as defined in [11]. 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, Veto, k-approval, Bucklin, and Borda, for both weighted and unweighted voters. For instance, 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 number of candidates is bounded by a small constant.

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, and provide initial results on the complexity of safe manipulation in these models.

In our first extension, a manipulator v may be followed by all voters who rank the same candidates above the current winner as v does. That is, in this model a voter u may follow v if any change of outcome that is beneficial to v is also beneficial to u. We show that some of the positive algorithmic results for the standard model also hold in this more general setting. In our second model, a voter u may follow a manipulator v that proposes to vote L, if, roughly, there are circumstances when voting L is beneficial to u. This model tends to be computationally more challenging: we show that finding a safe strategic vote in this setting is hard even for very simple voting rules.

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 considered in this paper. All of these rules assign scores to all candidates; the winner is then selected among the candidates with the highest score using a *tie-breaking rule*, i.e., a mapping $T : 2^C \to C$ that satisfies $T(S) \in S$. Unless specified otherwise, we assume that the tie-breaking rule is *lexicographic*, i.e., given a set of tied alternatives, it selects one that is maximal with respect to a fixed ordering $\succ$.

Given a vector $\alpha = (\alpha_1, \dots, \alpha_m)$ 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 an 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 $\lceil n/2 \rceil$.

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 $(R_1, \dots, R_n)$ under a voting rule $\mathcal{F}$ given voters' weights $\mathbf{w} = (w_1, \dots, w_n)$, we apply $\mathcal{F}$ on a modified profile which for each $i = 1, \dots, n$ 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 $\mathbf{w} = (1, \dots, 1)$, 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 $\mathcal{R} = (R_1, \dots, R_n)$.

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. Under truthful voting, a and b get 3 points, and c and d get 1 point each. Since ties are broken lexicographically, 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 two natural algorithmic questions. In the definitions below, $\mathcal{F}$ is a given voting rule and the voters are assumed to be unweighted.

- 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 $\mathcal{F}$?

The variants of these problems for weighted voters will be denoted, respectively, by wIsSafe($\mathcal{F}$) and wExistSafe($\mathcal{F}$). Note that, in general, it is not clear if an efficient algorithm for (w)ExistSafe($\mathcal{F}$) can be used to solve (w)IsSafe($\mathcal{F}$), or vice versa. However, if the number of candidates is constant, (w)ExistSafe($\mathcal{F}$) reduces to (w)IsSafe($\mathcal{F}$). We formulate the following two results for weighted voters; clearly, they also apply to unweighted voters.

Proposition 1. *Consider any voting rule $\mathcal{F}$. For any constant k, if $|C| \leq k$, then a polynomial-time algorithm for* wIsSafe($\mathcal{F}$) *can be used to solve* wExistSafe($\mathcal{F}$) *in 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 such rule, a polynomial-time algorithm for* wIsSafe($\mathcal{F}_\alpha$) *can be used to solve* wExistSafe($\mathcal{F}_\alpha$) *in polynomial time.*

Proof. We consider case (i); case (ii) is similar. There are at most $n^k = \mathrm{poly}(n)$ 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 wExistSafe($\mathcal{F}_\alpha$), it suffices to run wIsSafe($\mathcal{F}_\alpha$) on $\mathrm{poly}(n)$ 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.

Further, we note that for unweighted voters it is easy to check if a given manipulation is safe.

Proposition 3. *The problem* IsSafe($\mathcal{F}$) *is in* P *for any (anonymous) voting rule $\mathcal{F}$.*

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

Together with Propositions 1 and 2, Proposition 3 implies that the problem EXISTSAFE($\mathcal{F}$) is in P for Plurality, Veto, k-veto and k-approval for constant k, 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. However, it is not hard to show that the problem remains easy when all weights are small (polynomially bounded).

4 Plurality, veto, and k-approval

We will now show that the easiness results for k-approval and k-veto extend to arbitrary $k \le m$ and weighted voters (note that the distinction between k-veto and $(m - k)$-approval only matters for constant k).

Theorem 4. *For k-approval, the problems* wISSAFE *and* wEXISTSAFE *are in* P.

Proof. Fix a voter $v \in V$. To simplify notation, we renumber the candidates so that v's preference order is given by $c_1 \succ_v \dots \succ_v c_m$. 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 $i = 1, \dots, m$, 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 the winner in $\mathcal{R}_{-V_v}(L)$ is a candidate c_i with $i < j$.*

Proof. Suppose that L is a safe strategic vote for v. Then there exists an $i < j$ and a $U \subseteq V_v$ such that the winner in $\mathcal{R}_{-U}(L)$ is c_i. 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, the winner in $\mathcal{R}_{-V_v}(L)$ is not c_j; since L is safe, this means that the winner in $\mathcal{R}_{-V_v}(L)$ is c_ℓ for some $\ell < j$.

For the opposite direction, suppose that the winner in $\mathcal{R}_{-V_v}(L)$ is 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 $j \le k$, 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 c_j is the winner. If $s_i(\mathcal{R}_{-U}(L)) > s_j(\mathcal{R}_{-U}(L))$, then 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 WISSAFE: 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 WEXISTSAFE. We need to consider two cases.

$\mathbf{j > k}$**:**
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 $i = k+1, \dots, j-1$ 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.

$\mathbf{j \leq k}$**:**
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 $s_1, \dots, s_{j-1}$.

Set $C_g = \{c_1, \dots, c_{j-1}\}$, $C_b = \{c_j, \dots, c_m\}$. 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 $\ell = j, \dots, k$. 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 $\ell = 1, \dots, m$.

Indeed, if such a set C_{safe} exists, voter v can place the candidates in C_{safe} in positions $j, \dots, k$ 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 $L \neq R$ the winner in $\mathcal{R}_{-V_v}(L)$ is a candidate in C_b, and thus by Lemma 1 L is not safe. □

We remark that Theorem 4 crucially relies on the fact that we break ties based on a fixed priority ordering over the candidates. Indeed, it can be shown that there exists a (non-lexicographic) tie-breaking rule such that finding a safe vote with respect to k-approval combined with this tie-breaking rule is computationally hard (assuming k is viewed as a part of the input). As the focus of this paper is on lexicographic tie-breaking, we omit the formal statement and the proof of this fact.

In contrast, we can show that 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 5. WISSAFE($\mathcal{F}$) *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.*

Proof. For one candidate, the statement is trivial. With two candidates, every positional scoring rule is equivalent to Plurality, and under Plurality with two candidates no voter has an incentive to vote strategically.

Now, suppose that $|C| = 3$. Consider a voter i and assume without loss of generality that $R_i = (c_1, c_2, c_3)$. If $\mathcal{F}(\mathcal{R}) = c_1$, then i has no incentive to vote strategically. We will now consider the cases $\mathcal{F}(\mathcal{R}) = c_2$ and $\mathcal{F}(\mathcal{R}) = c_3$ separately.

1. $\mathcal{F}(\mathcal{R}) = c_2$. Suppose that L is a strategic vote for i. Then L cannot rank c_2 in top two positions. Indeed, any such manipulation does not decrease c_2's score and does not increase c_1's score. Thus, if c_2 had a higher score than c_1, this would still be the case no matter how many voters in V_i switch to voting L. Further, if both c_2 and c_1 had top scores, then L could succeed only if it does not change the scores of either of them. But in this case the score of c_3 does not change either, so the outcome remains the same. Thus, it remains to consider two cases: $L = (c_1, c_3, c_2)$ and $L = (c_3, c_1, c_2)$. Now, let $c = \mathcal{F}(\mathcal{R}_{-V_i}(L))$. If $c = c_3$, L is not safe. Further, if $c = c_2$, then we have $c_2 = \mathcal{F}(\mathcal{R}_{-U}(L))$ for any $U \subseteq V_i$, i.e., L is not a strategic vote for i. Finally, if $c = c_1$, then L is a safe strategic vote. Indeed, suppose that L is not safe, i.e., $\mathcal{F}(\mathcal{R}_{-U}(L)) = c_3$ for some $U \subset V_i$. Each switch from R_i to L does not decrease c_3's score, so in that case c_3 would be a winner in $\mathcal{R}_{-V_i}(L)$, a contradiction.

2. $\mathcal{F}(\mathcal{R}) = c_3$. It can be checked that if L is a strategic vote for i, then L has to rank c_2 first, i.e., $L \in \{(c_2, c_1, c_3), (c_2, c_3, c_1)\}$. If $\mathcal{F}(\mathcal{R}_{-V_i}(L)) = c_3$, by the same argument as above, there is no incentive for i to vote for L. Otherwise, L is a safe strategic vote, since c_3 is the least preferred candidate.

□

5 Bucklin and Borda

Bucklin rule is quite similar to k-approval, so we can use the ideas in the proof of Theorem 4 to design a polynomial-time algorithm for finding a safe manipulation with respect to Bucklin. However, the proof becomes significantly more complicated.

Theorem 6. *For the Bucklin rule,* wExistSafe *is in* P.

Interestingly, despite the intuition that wIsSafe should be easier than wExistSafe, it turns out that wIsSafe for Bucklin is coNP-hard.

Theorem 7. *For the Bucklin rule,* wIsSafe *is* coNP*-hard, even for a constant number of candidates.*

Proof. We give a reduction from Subset Sum. Recall that an instance of Subset Sum is given by a set of positive integers $A = \{a_1, \ldots, a_s\}$ and a positive integer K. It is a "yes"-instance if there is a subset of indices $I \subseteq \{1, \ldots, s\}$ such that $\sum_{i \in I} a_i = K$ and a "no"-instance otherwise. We assume without loss of generality that $K < \sum_{a_i \in A} a_i$.

Given an instance (A, K) of Subset Sum with $|A| = s$ and $\sum_{i=1}^{s} a_i = S$, we construct an instance of wIsSafe as follows. Set $C = \{a, b, c, x, y, z, x', y', z'\}$, and let $V = \{v_1, \ldots, v_s, u_1, u_2, u_3, u_4\}$. Table 1 shows the preferences and weights of each voter; observe that the total weight of all voters is $4S$. We ask if the vote $L = (a, c, b, x, y, z, x', y', z')$ is a safe strategic vote for v_1 under Bucklin; as we will see, the answer to this question does not depend on the tie-breaking rule.

Table 1:

Voter	Preference order	Weight
v_i	$(x, y, z, a, b, c, x', y', z')$	a_i
u_1	$(a, c, b, x, y, z, x', y', z')$	$2S - K - 1$
u_2	$(x, c, b, a, y, z, x', y', z')$	1
u_3	$(y, z, b, a, c, x, x', y', z')$	K
u_4	$(x', y', z', a, b, c, x, y, z)$	S

If all voters vote sincerely, then b wins in round 3 with $2S$ points, and all other voters get less that $2S$ points in the first three rounds. Note also that the total weight of voters in $C \setminus V_{v_1}$ that rank a first is $2S - K - 1$, and the total weight of voters in $C \setminus V_{v_1}$ that rank c second is $2S - K$.

Suppose that a group of voters $U \subseteq V_{v_1}$ votes L. If $w(U) < K$, then b remains the winner, while if $w(U) > K$ then a becomes the winner, as it gets the majority of votes in the first round. Therefore, L is a strategic vote for v_1. However, if $w(U) = K$, a only gets $2S - 1$ points in any of the first three rounds, while c gets $2S$ points in the second round. Therefore, in this case c wins, i.e., L is not safe for v_1. Hence, L is a safe strategic vote for v_1 if and only if no subset of A sums to K. □

For Borda, unlike k-approval and Bucklin, both of our problems are hard when the voters are weighted. The proof of the following theorem is similar to that of Theorem 7.

Theorem 8. *For the Borda rule,* wISSAFE *and* wEXISTSAFE *are* coNP*-hard. The hardness result holds even if there are only* 5 *candidates.*

6 Extensions of the Safe Strategic Voting Model

So far, 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 v and his would-be follower u rank the current winner last, it is easy to see that following v's recommendation that leads to displacing the current winner is in u'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 gets 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 be the winner under truthful voting. For any $v \in V$, let $I(v, c)$ denote the set of candidates that v ranks strictly above c. We say that two voters u and v are *similar* if $I(u, c) = I(v, c)$. A *similar set* S_v of a voter v for a given preference profile $\mathcal{R}$ and a voting rule $\mathcal{F}$ is given by $S_v = \{u \mid I(u, c) = I(v, c)\}$. (The set S_v depends on $\mathcal{R}$ and $\mathcal{F}$; however, for readability we omit $\mathcal{R}$ and $\mathcal{F}$ from the notation).

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

We can now adapt Definition 1 to our setting by replacing V_v with S_v.

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

Observe that if L is a (safe) strategic vote for v at $\mathcal{R}$ under $\mathcal{F}$, then it is also a (safe) strategic vote for any $u \in S_v$. Indeed, $u \in S_v$ implies $S_u = S_v$ and for any $a \in C$ we have $a \succ_u \mathcal{F}(\mathcal{R})$ if and only if $a \succ_v \mathcal{F}(\mathcal{R})$. Note also that we do not require $L \neq R_v$: indeed, in the preference-based extension $L = R_v$ may be a non-trivial manipulation, as it may induce voters in $S_v \setminus \{v\}$ to switch their preferences to R_v. That is, a voter may manipulate the election simply by asking other voters with similar preferences to vote like he does. Finally, it is easy to see that for any voter v, the set S_v of similar voters is easy to compute.

The two computational problems considered throughout this paper, i.e., the safety of a given manipulation and the existence of a safe manipulation remain relevant for the preference-based model. We will refer to these problems in this setting as IsSafe^{pr} and ExistSafe^{pr}, respectively, and use prefix W to denote their weighted variants. The problems (W)IsSafe^{pr} and (W)ExistSafe^{pr} appear to be somewhat harder than their counterparts in the original model. Indeed, while voters in S_v have similar preferences, their truthful votes may be substantially different, so it now matters *which* of the voters in S_v decide to follow the manipulator (rather than just *how many* of them, as in the original model). In particular, it is not clear if $\text{IsSafe}^{pr}(\mathcal{F})$ is polynomial-time solvable for any voting rule $\mathcal{F}$. However, it turns out that both of our problems are easy for k-approval, even with weighted voters.

Theorem 9. *For k-approval, the problems* wIsSafe^{pr} *and* wExistSafe^{pr} *are in* P.

In the preference-based model, a voter v follows a recommendation to vote in a particular way if it comes from a voter whose preferences are similar to those of v. 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:

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

Observe 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 (*); note also that we always have $i \in F_i(L)$.

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. We will denote the computational problems of checking whether a given vote is a safe strategic vote for a given voter in the goal-based extension and whether a given voter has a safe strategic vote in the goal-based extension by $\textsc{IsSafe}^{gl}$ and $\textsc{ExistSafe}^{gl}$, respectively, and use the prefix W to refer to weighted versions of these problems.

Two remarks are in order. First, it may be the case that even though i benefits from proposing to vote L, he is never pivotal with respect to (i, L) (this can happen, e.g., if i's weight is much smaller that that of the other voters). Thus, we need to explicitly include i in the set $F_i(L)$, to avoid the paradoxical situation where $i \notin F_i(L)$. Second, our definition of a safe vote only guarantees safety to the original manipulator, but not to her followers. In contrast, in the preference-based extension, any vote that is safe for the original manipulator is also safe for all similar voters.

The definition of a 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, and the ties are broken according to the priority 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 that 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 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 is beneficial 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 considerably harder to deal with 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 (with unweighted voters).

Theorem 10. *Given an election (C, V) with a preference profile $\mathcal{R}$ and unweighted voters, a manipulator i, and a vote L, we can compute the set $F_i(L)$ with respect to Plurality in time polynomial in the input size.*

We can use Theorem 10 to show that under Plurality one can determine in polynomial time whether a given vote L is safe for a player i, as well as find a safe strategic vote for i if one exists, as long as the voters are unweighted.

Theorem 11. *The problems* $\textsc{IsSafe}^{gl}$ *and* $\textsc{ExistSafe}^{gl}$ *are polynomial-time solvable for Plurality.*

For weighted voters, computing the follower set is computationally hard even for Plurality. While this result does not immediately imply that $\textsc{wIsSafe}^{gl}$ and $\textsc{wExistSafe}^{gl}$ are also hard for Plurality, it indicates that these problems are unlikely to be easily solvable.

Theorem 12. *Given an instance* $(C, V, \mathbf{w}, \mathcal{R})$ *of Plurality elections, voters* $i, j \in V$ *and a vote* L*, it is* NP*-hard to decide whether* $j \in F_i(L)$*.*

Just a little further afield, checking whether a given vote is safe with respect to 3-approval is computationally hard even for unweighted voters. This is in contrast with the standard model and the preference-based extension, where safely manipulating k-approval is easy for arbitrary k.

Theorem 13. $\textsc{IsSafe}^{gl}$ *is* coNP*-hard for* 3*-approval.*

Thus, while the preference-based extension appears to be similar to the original model of [11] from the computational perspective, the goal-based extension is considerably more difficult to work with.

7 Conclusions

In this paper, we started the investigation of algorithmic complexity of safe manipulation, as defined by Slinko and White [11]. We showed that finding a safe manipulation is easy for k-approval for an arbitrary value of k and for Bucklin, even with weighted voters. Somewhat surprisingly, checking whether a given manipulation is safe appears to be a more difficult problem, at least for weighted voters: while this problem is polynomial-time solvable for k-approval, it is coNP-hard for Bucklin. For the Borda rule, both checking whether a given manipulation is safe and identifying a safe manipulation is hard when the voters are weighted.

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. Our first extension of the model of [11] is very simple and natural, and seems to behave similarly to the original model from the algorithmic perspective. However, arguably, it does not capture some of the scenarios that may occur in practice. Our second model is considerably richer, but many of the associated computational problems become intractable.

A natural 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) for unweighted voters. 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 other types of tie-breaking rules, such as, e.g., 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.

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, W. H. Freeman, 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, Addison Wesley, 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. Non-dictatorial Social Choice Rules Are Safely Manipulable, 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.

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

Edith Elkind
Division of Mathematical Science
Nanyang Technological University
Singapore
Email: `eelkind@ntu.edu.sg`