

Detecting Possible Manipulators in Elections

Palash Dey
Department of Computer
Science and Automation
Indian Institute of Science,
Bangalore, India.
palash@csa.iisc.ernet.in

Neeldhara Misra
Department of Computer
Science and Automation
Indian Institute of Science,
Bangalore, India.
mail@neeldhara.com

Y. Narahari
Department of Computer
Science and Automation
Indian Institute of Science,
Bangalore, India.
hari@csa.iisc.ernet.in

ABSTRACT

Manipulation is a problem of fundamental importance in the context of voting in which the voters exercise their votes strategically instead of voting honestly to prevent selection of an alternative that is less preferred. The Gibbard-Satterthwaite theorem shows that there is no strategy-proof voting rule that simultaneously satisfies certain combinations of desirable properties. Researchers have attempted to get around the impossibility results in several ways such as domain restriction and computational hardness of manipulation. However these approaches have been shown to have limitations. Since prevention of manipulation seems to be elusive, an interesting research direction therefore is detection of manipulation. Motivated by this, we initiate the study of detection of possible manipulators in an election.

We formulate two pertinent computational problems - Coalitional Possible Manipulators (CPM) and Coalitional Possible Manipulators given Winner (CPMW), where a suspect group of voters is provided as input to compute whether they can be a potential coalition of possible manipulators. In the absence of any suspect group, we formulate two more computational problems namely Coalitional Possible Manipulators Search (CPMS), and Coalitional Possible Manipulators Search given Winner (CPMSW). We provide polynomial time algorithms for these problems, for several popular voting rules. For a few other voting rules, we show that these problems are in NP-complete. We observe that detecting manipulation maybe easy even when manipulation is hard, as seen for example, in the case of the Borda voting rule.

Categories and Subject Descriptors

F.2 [Theory of Computation]: Analysis of Algorithms and Problem Complexity;

I.2.11 [Artificial Intelligence]: Distributed Artificial Intelligence—Multiagent Systems

General Terms

Algorithms, Theory

Appears in: *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015)*, Bordini, Elkind, Weiss, Yolum (eds.), May 4–8, 2015, Istanbul, Turkey.
Copyright © 2015, International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

Keywords

Computational social choice; voting; manipulation; detection

1. INTRODUCTION

On many occasions, agents need to agree upon a common decision although they have different preferences over the available alternatives. A natural approach used in these situations is voting. Some classic examples of the use of voting rules in the context of multiagent systems are in collaborative filtering [28], rank aggregation in web [8] etc.

In a typical voting scenario, we have a set of m candidates and a set of n voters reporting their rankings of the candidates called their preferences or votes. A voting rule selects one candidate as the winner once all the voters provide their votes. A set of votes over a set of candidates along with a voting rule is called an election. A basic problem with voting rules is that the voters may vote strategically instead of voting honestly, leading to the selection of a candidate which is not the *actual winner*. We call a candidate actual winner if, it wins the election when every voter votes truthfully. This phenomenon of strategic voting is called *manipulation* in the context of voting. The Gibbard-Satterthwaite (G-S) theorem [20, 31] says that manipulation is unavoidable for any *unanimous* and *non-dictatorial* voting rule if we have at least three candidates. A voting rule is called *unanimous* if whenever any candidate is most preferred by all the voters, such a candidate is the winner. A voting rule is called *non-dictatorial* if there does not exist any voter whose most preferred candidate is always the winner irrespective of the votes of other voters. The problem of manipulation is particularly relevant for multiagent systems since agents have computational power to determine strategic votes. There have been several attempts to bypass the impossibility result of the G-S theorem.

Economists have proposed domain restriction as a way out of the impossibility implications of the G-S theorem. The G-S theorem assumes all possible preference profiles as the domain of voting rules. In a restricted domain, it has been shown that we can have voting rules that are not vulnerable to manipulation. A prominent restricted domain is the domain of single peaked preferences, in which the median voting rule provides a satisfactory solution [22]. To know more about other domain restrictions, we refer to [22, 17]. This approach of restricting the domain, however, suffers from the requirement that the social planner needs to know the domain of preference profiles of the voters, which is often impractical.

1.1 Related Work

Researchers in computational social choice theory have proposed invoking computational intractability of manipulation as a possible work around for the G-S theorem. Bartholdi et al. [1, 2] first proposed the idea of using computational hardness as a barrier against manipulation. Bartholdi et al. defined and studied the computational problem called manipulation where a set of manipulators have to compute their votes that make their preferred candidate win the election. The manipulators know the votes of the truthful voters and the voting rule that will be used to compute the winner. Following this, a large body of research [25, 6, 39, 38, 5, 27, 10, 13, 24, 19, 26, 12, 40, 7, 15, 9] shows that the manipulation problem is in NP-complete (NPC) for many voting rules. However, Procaccia et al. [29, 30] showed average case easiness of manipulation assuming *junta* distribution over the voting profiles. Friedgut et al. [16] showed that any *neutral* voting rule which is sufficiently far from being dictatorial is manipulable with non-negligible probability at any uniformly random preference profile by a uniformly random preference. The above result holds for elections with three candidates only. A voting rule is called *neutral* if the names of the candidates are immaterial. Isaksson et al. [21] generalize the above result to any number of candidates. Walsh [32] empirically shows ease of manipulating an STV (single transferable vote) election – one of the very few voting rules where manipulation even by one voter is in NPC [1]. In addition to the results mentioned above, there exist many other results in the literature that emphasize the weakness of considering computational complexity as a barrier [4, 37, 36, 14, 33]. Hence, the barrier of computational hardness is ineffective against manipulation in many settings.

1.2 Motivation

In a situation where multiple attempts for prevention of manipulation fail to provide a fully satisfactory solution, detection of manipulation is a natural next step of research. There have been scenarios where establishing the occurrence of manipulation is straightforward, by observation or hindsight. For example, in sport, there have been occasions where the very structure of the rules of the game have encouraged teams to deliberately lose their matches. Observing such occurrences in, for example, football (the 1982 FIFA World Cup football match played between West Germany and Austria) and badminton (the quarter-final match between South Korea and China in the London 2012 Olympics), the relevant authorities have subsequently either changed the rules of the game (as with football) or disqualified the teams in question (as with the badminton example). The importance of detecting manipulation lies in the potential for implementing corrective measures in the future. For reasons that will be evident soon, it is not easy to formally define the notion of manipulation detection. Assume that we have the votes from an election that has already happened. A voter is potentially a manipulator if there exists a preference \succ , different from the voter’s reported preference, which is such that the voter had an “incentive to deviate” from the former. Specifically, suppose the candidate who wins with respect to this voter’s reported preference is preferred (in \succ) over the candidate who wins with respect to \succ . In such a situation, \succ could potentially be the voter’s truthful preference, and the voter could be refraining from being truthful because an untruthful vote leads to a more

favorable outcome with respect to \succ . Note that we do not (and indeed, cannot) conclusively suggest that a particular voter has manipulated an election. This is because the said voter can always claim that she voted truthfully; since her actual preference is only known to her, there is no way to prove or disprove such a claim. Therefore, we are inevitably limited to asking only whether or not a voter has *possibly* manipulated an election.

Despite this technical caveat, it is clear that efficient detection of manipulation, even if it is only possible manipulation, is potentially of interest in practice. We believe that, the information whether a certain group of voters have possibly manipulated an election or not would be very useful to social planners. For example, the organizers of an event, say London 2012 Olympics, maybe very interested to have this information. Also, in settings where data from many past elections (roughly over a fixed set of voters) is readily available, it is conceivable that possible manipulation could serve as suggestive evidence of real manipulation. Aggregate data about possible manipulations, although formally inconclusive, could serve as an important evidence of real manipulation, especially in situations where the instances of possible manipulation turn out to be statistically significant. Thus, efficient detection of possible manipulation would provide a very useful input to a social planner for future elections. We remark that having a rich history is typically not a problem, particularly for AI related applications, since the data generated from an election is normally kept for future requirements (for instance, for data mining or learning). For example, several past affirmatives for possible manipulation is one possible way of formalizing the notion of erratic past behavior. Also, applications where benefit of doubt maybe important, for example, elections in judiciary systems, possible manipulation detection seems useful. Thus the computational problem of detecting possible manipulation is of definite interest in this setting.

1.3 Contributions

The novelty of this paper is in initiating research on *detection of possible manipulators* in elections. We formulate four pertinent computational problems in this context:

- CPM: In the *coalitional possible manipulators* problem, we are interested in whether or not a given subset of voters is a possible coalition of manipulators [Definition 4].
- CPMW: The *coalitional possible manipulators given winner* is the CPM problem with the additional information about who the winner would have been if the possible manipulators had all voted truthfully [Definition 2].
- CPMS, CPMSW: In CPMS (*Coalitional Possible Manipulators Search*), we want to know, whether there exists any coalition of possible manipulators of a size at most k [Definition 6]. Similarly, we define CPMSW (*Coalitional Possible Manipulators Search given Winner*) [Definition 5].

Our specific findings are as follows.

- We show that all the four problems above, for scoring rules and the maximin voting rule, are in P when the coalition size is one [Theorem 1 and Theorem 4].

- We show that all the four problems, for any coalition size, are in P for a wide class of scoring rules which include the Borda voting rule [Theorem 2, Theorem 3 and Corollary 1].
- We show that, for the Bucklin voting rule [Theorem 6], both the CPM and CPMW problems are in P, for any coalition size. The CPMS and CPMSW problems for the Bucklin voting rule are also in P, when we have maximum possible coalition size $k = O(1)$.
- We show that both the CPM and the CPMW problems are in NPC for the STV voting rule [Theorem 7 and Corollary 2], even for a coalition of size one. We also prove that the CPMW problem is in NPC for maximin voting rule [Theorem 5], for a coalition of size two.

We observe that all the four problems are computationally easy for many voting rules that we study in this paper. This can be taken as a positive result. The results for the CPM and the CPMW problems are summarized in Table 1.

| Voting Rule | CPM, $k = 1$ | CPM | CPMW, $k = 1$ | CPMW |
|---------------|--------------|-----|---------------|------|
| Scoring Rules | P | ? | P | ? |
| Borda | P | P | P | P |
| k -approval | P | P | P | P |
| Maximin | P | ? | P | NPC |
| Bucklin | P | P | P | P |
| STV | NPC | NPC | NPC | NPC |

Table 1: Results for CPM and CPMW (k denotes coalition size). The ‘?’ mark means that the problem is open.

Organization.

The rest of the paper is organized as follows. We describe the necessary preliminaries in Section 2; we formally define the computational problems in Section 3; we present the results in Section 4 and finally we conclude in Section 5.

2. PRELIMINARIES

Let $\mathcal{V} = \{v_1, \dots, v_n\}$ be the set of all voters and $\mathcal{C} = \{c_1, \dots, c_m\}$ the set of all candidates. If not mentioned otherwise, n and m denote the number of voters and the number of candidates respectively. Each voter v_i 's vote is a preference \succ_i over the candidates which is a linear order over \mathcal{C} . For example, for two candidates a and b , $a \succ_i b$ means that the voter v_i prefers a to b . We will use $a \succ_i b$ to denote the fact that $a \succ_i b$, $a \neq b$. We denote the set of all linear orders over \mathcal{C} by $\mathcal{L}(\mathcal{C})$. Hence, $\mathcal{L}(\mathcal{C})^n$ denotes the set of all n -voters' preference profile $(\succ_1, \dots, \succ_n)$. We denote the $(n - 1)$ -voters' preference profile by $(\succ_1, \dots, \succ_{i-1}, \succ_{i+1}, \dots, \succ_n)$ by \succ_{-i} . We denote the set $\{1, 2, 3, \dots\}$ by \mathbb{N}^+ . The power set of \mathcal{C} is denoted by $2^{\mathcal{C}}$, and \emptyset denotes the empty set. A map $r_c : \cup_{n, |C| \in \mathbb{N}^+} \mathcal{L}(\mathcal{C})^n \rightarrow 2^{\mathcal{C}} \setminus \{\emptyset\}$ is called a voting correspondence. A map $t : \cup_{|C| \in \mathbb{N}^+} 2^{\mathcal{C}} \setminus \{\emptyset\} \rightarrow \mathcal{C}$ is called a tie breaking rule. Commonly used tie breaking rules are lexicographic tie breaking rules where ties are broken according to a predetermined preference $\succ_{-i} \in \mathcal{L}(\mathcal{C})$. A voting rule is $r = t \circ r_c$, where \circ denotes composition of mappings. Given an election E , we can construct a weighted graph G_E called weighted majority graph from E . The set of vertices in G_E is the set of candidates in E . For any two candidates x and y , the weight on the edge (x, y) is $D_E(x, y) =$

$N_E(x, y) - N_E(y, x)$, where $N_E(x, y)$ (respectively $N_E(y, x)$) is the number of voters who prefer x to y (respectively y to x). Some examples of common voting correspondences are as follows.

- Positional scoring rules: Given an m -dimensional vector $\vec{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_m) \in \mathbb{R}^m$ with $\alpha_1 \geq \alpha_2 \geq \dots \geq \alpha_m$ and $\alpha_1 > \alpha_m$, we can naturally define a voting rule - a candidate gets score α_i from a vote if it is placed at the i^{th} position, and the score of a candidate is the sum of the scores it receives from all the votes. The winners are the candidates with maximum score. For $\alpha = (m - 1, m - 2, \dots, 1, 0)$, we get the Borda voting rule. With $\alpha_i = 1 \forall i \leq k$ and 0 else, the voting rule we get is known as the k -approval voting rule. The plurality voting rule is the 1-approval voting rule and the veto voting rule is the $(m - 1)$ -approval voting rule.
- Maximin: The maximin score of a candidate x is $\min_{y \neq x} D(x, y)$. The winners are the candidates with maximum maximin score.
- Bucklin: A candidate x 's Bucklin score is the minimum number l such that more than half of the voters rank x in their top l positions. The winners are the candidates with lowest Bucklin score. This voting rule is also sometimes referred as the simplified Bucklin voting rule.
- Single Transferable Vote: In Single Transferable Vote (STV), a candidate with least plurality score is dropped out of the election and its votes are transferred to the next preferred candidate. If two or more candidates receive least plurality score, then some predetermined tie breaking rule is used. The candidate that remains after $(m - 1)$ rounds is the winner.

3. PROBLEM FORMULATION

Consider an election that has already happened in which all the votes are known and thus the winner $x \in \mathcal{C}$ is also known. We call the candidate x the current winner of the election. The authority may suspect that the voters belonging to $M \subset \mathcal{V}$ have formed a coalition among themselves and manipulated the election by voting non-truthfully. The authority believes that other voters who do not belong to M , have voted truthfully. We denote $|M|$, the size of the coalition, by k . Suppose the authority has auxiliary information, maybe from some other sources, which says that the actual winner should have been some candidate $y \in \mathcal{C}$ other than x . We call a candidate actual winner if it wins the election where all the voters vote truthfully. This means that the authority thinks that, had the voters in M voted truthfully, the candidate y would have been the winner. We remark that there are practical situations, for example, 1982 FIFA World cup or 2012 London Olympics, where the authority knows the actual winner. This situation is formalized below.

Definition 1. Let r be a voting rule, and $(\succ_i)_{i \in \mathcal{V}}$ be a voting profile of a set \mathcal{V} of n voters. Let x be the winning candidate with respect to r for this profile. For a candidate $y \neq x$, $M \subset \mathcal{V}$ is said to be a coalition of possible manipulators against y with respect to r if there exists a $|M|$ -voters' profile $(\succ'_j)_{j \in M} \in \mathcal{L}(\mathcal{C})^{|M|}$ such that $x \succ'_j y, \forall j \in M$, and further, $r((\succ'_j)_{j \in \mathcal{V} \setminus M}, (\succ'_i)_{i \in M}) = y$.

Using the notion of coalition of possible manipulators, we formulate a computational problem called *Coalitional Possible Manipulators given Winner (CPMW)* as follows.

Definition 2. (CPMW Problem)

Given a voting rule r , a preference profile $(\succ_i)_{i \in \mathcal{V}}$ of a set of voters \mathcal{V} over a set of candidates \mathcal{C} , a subset of voters $M \subset \mathcal{V}$, and a candidate y , determine if M is a coalition of possible manipulators against y with respect to r .

In the CPMW problem, the actual winner is given in the input. However, it may very well happen that the authority does not have any other information to guess the *actual winner* - the candidate who would have won the election had the voters in M voted truthfully. In this situation, the authority is interested in knowing whether there is a $|M|$ -voter profile which along with the votes in $\mathcal{V} \setminus M$ makes some candidate $y \in \mathcal{C}$ the winner who is different from the current winner $x \in \mathcal{C}$ and all the preferences in the $|M|$ -voters' profile prefer x to y . If such a $|M|$ -voter profile exists for the subset of voters M , then we call M a *coalition of possible manipulators* and the corresponding computational problem is called *Coalitional Possible Manipulators (CPM)*. These notions are formalized below.

Definition 3. Let r be a voting rule, and $(\succ_i)_{i \in \mathcal{V}}$ be a voting profile of a set \mathcal{V} of n voters. A subset of voters $M \subset \mathcal{V}$ is called a *coalition of possible manipulators with respect to r* if M is a coalition of possible manipulators against some candidate y with respect to r .

Definition 4. (CPM Problem)

Given a voting rule r , a preference profile $(\succ_i)_{i \in \mathcal{V}}$ of a set of voters \mathcal{V} over a set of candidates \mathcal{C} , and a subset of voters $M \subset \mathcal{V}$, determine if M is a coalition of possible manipulators with respect to r .

In both the CPMW and CPM problems, a subset of voters which the authority suspect to be a coalition of manipulators, is given in the input. However, there can be situations where there is no specific subset of voters to suspect. In those scenarios, it may still be useful to know, what are the possible coalition of manipulators of size less than some number k . Towards that end, we extend the CPMW and CPM problems to search for a coalition of potential possible manipulators and call them *Coalitional Possible Manipulators Search given Winner (CPMSW)* and *Coalitional Possible Manipulators Search (CPMS)* respective.

Definition 5. (CPMSW Problem)

Given a voting rule r , a preference profile $(\succ_i)_{i \in \mathcal{V}}$ of a set of voters \mathcal{V} over a set of candidates \mathcal{C} , a candidate y , and an integer k , determine whether there exists any $M \subset \mathcal{V}$ with $|M| \leq k$ such that M is a coalition of possible manipulators against y .

Definition 6. (CPMS Problem)

Given a voting rule r , a preference profile $(\succ_i)_{i \in \mathcal{V}}$ of a set of voters \mathcal{V} over a set of candidates \mathcal{C} , and an integer k , determine whether there exists any $M \subset \mathcal{V}$ with $|M| \leq k$ such that M is a coalition of possible manipulators.

3.1 Discussion

The CPMW problem may look very similar to the manipulation problem [2, 5]- in both the problems a set of voters

try to make a candidate winner. However, in the CPMW problem, the actual winner must be less preferred to the current winner. Although it may look like a subtle difference, it changes the nature and complexity theoretic behavior of the problem completely. For example, we show that all the four problems have an efficient algorithm for a large class of voting rules that includes the Borda voting rule, for any coalition size. However, for the manipulation problem for the Borda voting rule is in NPC, even when we have at least two manipulators [6, 3]. Another important difference is that the manipulation problem, in contrast to the problems studied in this paper, does not take care of manipulators' preferences. We believe that there does not exist any formal reduction between the CPMW problem and the manipulation problem.

On the other hand, the CPMS problem is similar to the margin of victory problem defined by Xia [34], where also we are looking for changing the current winner by changing at most some k number of votes, which in turn identical to the destructive bribery problem [11]. Whereas, in the CPMS problem, the vote changes can occur in a restricted fashion. An important difference between the two problems is that the margin of victory problem has the hereditary property which the CPMS problem does not possess (there is no coalition of possible manipulators of size n in any election for all the common voting rules). These two problems do not seem to have any obvious complexity theoretic implications.

Now, we explore the connections among the four problems that we study here. Notice that, a polynomial time algorithm for the CPM and the CPMW problems gives us a polynomial time algorithm for the CPMS and the CPMSW problems for any maximum possible coalition size $k = O(1)$. Also, observe that, a polynomial time algorithm for the CPMW (respectively CPMSW) problem implies a polynomial time algorithm for the CPM (respectively CPMS) problem. Hence, we have the following propositions.

Proposition 1. For every voting rule, if the maximum possible coalition size $k = O(1)$, then,

$$CPMW \in P \Rightarrow CPM, CPMSW, CPMS \in P$$

Proposition 2. For every voting rule,

$$CPMSW \in P \Rightarrow CPMS \in P$$

4. RESULTS

In this section, we present our algorithmic results for the CPMW, CPM, CPMSW, and CPMS problems for various voting rules.

4.1 Scoring Rules

Below we have certain lemmas which form a crucial ingredient of our algorithms. To begin with, we define the notion of a *manipulated preference*. Let r be a scoring rule and $\succ := (\succ_i, \succ_{-i})$ be a voting profile of n voters. Let \succ'_i be a preference such that

$$r(\succ) \succ'_i r(\succ'_i, \succ_{-i}).$$

Then, we say that \succ'_i is a (\succ, i) -manipulated preference with respect to r . We omit the reference to r if it is clear from the context.

Lemma 1. Let r be a scoring rule and $\succ := (\succ_i, \succ_{-i})$ be a voting profile of n voters. Let a and b be two candidates

such that $score_{\succ_{-i}}(a) > score_{\succ_{-i}}(b)$, and let \succ'_i be (\succ, i) -manipulated preference where a precedes b :

$$\succ'_i := \dots > a > \dots > b > \dots$$

If a, b are not winners with respect to either (\succ'_i, \succ_{-i}) or \succ , then the preference \succ''_i obtained from \succ'_i by interchanging a and b is also (\succ, i) -manipulated.

Proof: Let $x := r(\succ'_i, \succ_{-i})$. It suffices to show that x continues to win in the proposed profile (\succ''_i, \succ_{-i}) . To this end, it is enough to argue the scores of a and b with respect to x . First, consider the score of b in the new profile:

$$\begin{aligned} score_{(\succ''_i, \succ_{-i})}(b) &= score_{\succ''_i}(b) + score_{\succ_{-i}}(b) \\ &< score_{\succ'_i}(a) + score_{\succ_{-i}}(a) \\ &= score_{(\succ'_i, \succ_{-i})}(a) \\ &\leq score_{(\succ'_i, \succ_{-i})}(x) \\ &= score_{(\succ''_i, \succ_{-i})}(x) \end{aligned}$$

The second line uses the fact that $score_{\succ''_i}(b) = score_{\succ'_i}(a)$ and $score_{\succ_{-i}}(b) < score_{\succ_{-i}}(a)$. The fourth line comes from the fact that x is the winner and the last line follows from the fact that the position of x is same in both profiles. Similarly, we have the following argument for the score of a in the new profile (the second line below simply follows from the definition of scoring rules).

$$\begin{aligned} score_{(\succ''_i, \succ_{-i})}(a) &= score_{\succ''_i}(a) + score_{\succ_{-i}}(a) \\ &\leq score_{\succ'_i}(a) + score_{\succ_{-i}}(a) \\ &= score_{(\succ'_i, \succ_{-i})}(a) \\ &\leq score_{(\succ'_i, \succ_{-i})}(x) \\ &= score_{(\succ''_i, \succ_{-i})}(x) \end{aligned}$$

Since the tie breaking rule is according to some predefined fixed order $\succ_t \in \mathcal{L}(\mathcal{C})$ and the candidates tied with winner in (\succ''_i, \succ_{-i}) also tied with winner in (\succ'_i, \succ_{-i}) , we have the following,

$$r(\succ) \succ'_i r(\succ''_i, \succ_{-i})$$

□

We now show that, if there is some (\succ, i) -manipulated preference with respect to a scoring rule r , then there exists a (\succ, i) -manipulated preference with a specific structure.

Lemma 2. *Let r be a scoring rule and $\succ := (\succ_i, \succ_{-i})$ be a voting profile of n voters. If there is some (\succ, i) -manipulated preference with respect to r , then there also exists a (\succ, i) -manipulated preference \succ'_i where the actual winner y immediately follows the current winner x :*

$$\succ'_i := \dots > x > y > \dots$$

and the remaining candidates are in nondecreasing order of their scores from \succ_{-i} .

Proof: Let \succ'' be a (\succ, i) -manipulated preference with respect to r . Let $x := r(\succ), y := r(\succ'', \succ_{-i})$. From Lemma 1, without loss of generality, we may assume that, all candidates except x, y are in nondecreasing order of $score_{\succ_{-i}}(\cdot)$ in the preference \succ'' . If $\succ'' := \dots \succ x \succ \dots \succ y \succ \dots \succ \dots$, we define $\succ'_i := \dots \succ x \succ y \succ \dots \succ \dots$ from \succ''_i where y is moved to the position following x and the position of

the candidates in between x and y in \succ''_i is deteriorated by one position each. The position of the rest of the candidates remain same in both \succ''_i and \succ'_i . Now we have following,

$$\begin{aligned} score_{(\succ'_i, \succ_{-i})}(y) &= score_{\succ'_i}(y) + score_{\succ_{-i}}(y) \\ &\geq score_{\succ''_i}(y) + score_{\succ_{-i}}(y) \\ &= score_{(\succ''_i, \succ_{-i})}(y) \end{aligned}$$

We also have,

$$score_{(\succ'_i, \succ_{-i})}(a) \leq score_{(\succ''_i, \succ_{-i})}(a), \forall a \in \mathcal{C} \setminus \{y\}$$

Since the tie breaking rule is according to some predefined order $\succ_t \in \mathcal{L}(\mathcal{C})$, we have the following,

$$r(\succ) \succ'_i r(\succ'', \succ_{-i})$$

□

Using Lemma 1 and 2, we now present the results for the scoring rules.

Theorem 1. *The CPMW, CPM, CPMSW, and CPMS problems for scoring rules are in P for a coalition of size 1 (that is, $k = 1$).*

Proof: From Proposition 1, it is enough to give a polynomial time algorithm for the CPMW problem. So consider the CPMW problem. We are given the actual winner y and we compute the current winner x with respect to r . Let $\succ_{[j]}$ be a preference where x and y are in positions j and $(j + 1)$ respectively, and the rest of the candidates are in nondecreasing order of the score that they receive from \succ_{-i} . For $j \in \{1, 2, \dots, m - 1\}$, we check if y wins with the profile $(\succ_{-i}, \succ_{[j]})$. If we are successful with at least one j we report YES, otherwise we say NO. The correctness follows from Lemma 2. Thus we have a polynomial time algorithm for CPMW when $k = 1$. □

Now, we study the CPMW and the CPM problems when $k > 1$. If $m = O(1)$, then both the CPMW and the CPM problems for any anonymous and efficient voting rule r can be solved in polynomial time by iterating over all possible $\binom{m+k-1}{m!}$ ways the manipulators can have actual preferences. A voting rule is called efficient if winner determination under it is in P, for any coalition size.

Theorem 2. *For scoring rules with $\alpha_1 - \alpha_2 \leq \alpha_i - \alpha_{i+1}, \forall i$, the CPMW and the CPM problems are in P, for any coalition size.*

Proof: We provide a polynomial time algorithm for the CPMW problem in this setting. Let x be the current winner and y be the given actual winner. Let M be the given subset of voters. Let $((\succ_i)_{i \in M}, (\succ_j)_{j \in V \setminus M})$ be the reported preference profile. Without loss of generality, we assume that x is the most preferred candidate in every $\succ_i, i \in M$. Let us define $\succ'_i, i \in M$, by moving y to the second position in the preference \succ_i . In the profile $((\succ'_i)_{i \in M}, (\succ_j)_{j \in V \setminus M})$, the winner is either x or y since only y 's score has increased. We claim that M is a coalition of possible manipulators with respect to y if and only if y is the winner in preference profile $((\succ'_i)_{i \in M}, (\succ_j)_{j \in V \setminus M})$. This can be seen as follows. Suppose there exist preferences \succ''_i , with $x \succ''_i y, i \in M$, for which y wins in the profile $((\succ''_i)_{i \in M}, (\succ_j)_{j \in V \setminus M})$. Now without loss of generality, we

assume that y immediately follows x in all $\succ''_i, i \in M$, and $\alpha_1 - \alpha_2 \leq \alpha_i - \alpha_{i+1}, \forall i$ implies that we can also assume that x and y are in the first and second positions respectively in all $\succ''_i, i \in M$. Now in both the profiles, $((\succ'_i)_{i \in M}, (\succ_j)_{j \in V \setminus M})$ and $((\succ''_i)_{i \in M}, (\succ_j)_{j \in V \setminus M})$, the score of x and y are same. But in the first profile x wins and in the second profile y wins, which is a contradiction. \square

Theorem 3. *For scoring rules with $\alpha_1 - \alpha_2 \leq \alpha_i - \alpha_{i+1}, \forall i$, the CPMSW and the CPMS problems are in P, for any coalition size.*

Proof: From Proposition 2, it is enough to prove that $CPMSW \in \mathcal{P}$. Let x be the current winner, y be the given actual winner and $s(x)$ and $s(y)$ be their current respective scores. For each vote $v \in \mathcal{V}$, we compute a number $\Delta(v) = \alpha_2 - \alpha_j - \alpha_1 + \alpha_i$, where x and y are receiving scores α_i and α_j respectively from the vote v . Now, we output yes iff there are k votes $v_i, 1 \leq i \leq k$ such that, $\sum_{i=1}^k \Delta(v_i) \geq s(x) - s(y)$, which can be checked easily by sorting the $\Delta(v)$'s in nonincreasing order and checking the condition for the first k $\Delta(v)$'s, where k is the maximum possible coalition size specified in the input. The proof of correctness follows by exactly in the same line of argument as the proof of Theorem 2. \square

For the plurality voting rule, we can solve all the problems easily using max flow. Hence, from Theorem 2 and Theorem 3, we have the following result.

Corollary 1. *The CPMW, CPM, CPMSW, and CPMS problems for the Borda and k -approval voting rules are in P, for any coalition size.*

4.2 Maximin Voting Rule

For the maximin voting rule, we show that all the four problems are in P, when we have a coalition of size one.

Theorem 4. *The CPMW, CPM, CPMSW, and CPMS problems for maximin voting rule are in P for coalition size $k = 1$ or maximum possible coalition size $k = 1$.*

Proof: Given a n -voters' profile $\succ \in \mathcal{L}(\mathcal{C})^n$ and a voter v_i , let the current winner be $x := r(\succ)$ and the given actual winner be y . We will construct $\succ' = (\succ'_i, \succ_{-i})$, if it exists, such that $r(\succ) \succ'_i r(\succ') = y$, thus deciding whether v_i is a possible manipulator or not. Now, the maximin score of x and y in the profile \succ' can take one of values from the set $\{score_{\succ_{-i}}(x) \pm 1\}$ and $\{score_{\succ_{-i}}(y) \pm 1\}$. The algorithm is as follows. We first guess the maximin score of x and y in the profile \succ' . There are only four possible guesses. Suppose, we guessed that x 's score will decrease by one and y 's score will decrease by one assuming that this guess makes y win. Now notice that, without loss of generality, we can assume that y immediately follows x in the preference \succ'_i since y is the winner in the profile \succ' . This implies that there are only $O(m)$ many possible positions for x and y in \succ'_i . We guess the position of x and thus the position of y in \succ'_i . Let $B(x)$ and $B(y)$ be the sets of candidates with whom x and respectively y performs worst. Now since, x 's score will decrease and y 's score will decrease, we have the following constraint on \succ'_i . There must be a candidate each from $B(y)$ and $B(x)$ that will precede x . We do not know a-priori if there is one candidate

that will serve as a witness for both $B(x)$ and $B(y)$, or if there separate witnesses. In the latter situation, we also do not know what order they appear in. Therefore we guess if there is a common candidate, and if not, we guess the relative ordering of the distinct candidates from $B(x)$ and $B(y)$. Now we place any candidate at the top position of \succ'_i if this action does not make y lose the election. If there are many choices, we prioritize in favor of candidates from $B(x)$ and $B(y)$ — in particular, we focus on the candidates common to $B(x)$ and $B(y)$ if we expect to have a common witness, otherwise, we favor a candidate from one of the sets according to the guess we start with. If still there are multiple choices, we pick arbitrarily. After that we move on to the next position, and do the same thing (except we stop prioritizing explicitly for $B(x)$ and $B(y)$ once we have at least one witness from each set). The other situations can be handled similarly with minor modifications. In this way, if it is able to get a complete preference, then it checks whether v_i is a possible manipulator or not using this preference. If yes, then it returns YES. Otherwise, it tries other positions for x and y and other possible scores of x and y . After trying all possible guesses, if it cannot find the desired preference, then it outputs NO. Since there are only polynomial many possible guesses, this algorithm runs in a polynomial amount of time. The proof of correctness follows from the proof of Theorem 1 in [2]. \square

We now show that the CPMW problem for maximin voting rule is in NPC when we have $k > 1$. Towards that, we use the fact that the *unweighted coalitional manipulation (UCM)* problem for maximin voting rule is in NPC [39], when we have $k > 1$. The UCM problem is as follows.

Definition 7. (*UCM Problem*)

Given a voting rule r , a set of manipulators $M \subset \mathcal{V}$, a profile of non-manipulators' vote $(\succ_i)_{i \in \mathcal{V} \setminus M}$, and a candidate $z \in \mathcal{C}$, we are asked whether there exists a profile of manipulators' votes $(\succ'_j)_{j \in M}$ such that $r((\succ_i)_{i \in \mathcal{V} \setminus M}, (\succ'_j)_{j \in M}) = z$. Assume that ties are broken in favor of z .

We define a restricted version of the UCM problem called R-UCM as follows.

Definition 8. (*R-UCM Problem*)

This problem is the same as the UCM problem with a given guarantee - let $k := |M|$. The candidate z loses pairwise election with every other candidate by $4k$ votes. For any two candidates $a, b \in \mathcal{C}$, either a and b ties or one wins pairwise election against the other one by margin of either $2k + 2$ or of $4k$ or of $8k$. We denote the margin by which a candidate a defeats b , by $d(a, b)$.

The R-UCM problem for maximin voting rule is in NPC [39], when we have $k > 1$. We will need the following lemma to manipulate the pairwise difference scores in the reduction. The lemma has been used before [23, 35].

Lemma 3. *For any function $f : \mathcal{C} \times \mathcal{C} \rightarrow \mathbb{Z}$, such that*

1. $\forall a, b \in \mathcal{C}, f(a, b) = -f(b, a)$.

2. $\forall a, b \in \mathcal{C}, f(a, b)$ is even,

there exists a n voters' profile such that for all $a, b \in \mathcal{C}$, a defeats b with a margin of $f(a, b)$. Moreover,

$$n = O \left(\sum_{\{a, b\} \in \mathcal{C} \times \mathcal{C}} |f(a, b)| \right)$$

Theorem 5. *The CPMW problem for maximin voting rule is in NPC, for a coalition of size at least 2.*

Proof: Clearly the CPMW problem for maximin voting rule is in NP. We provide a many-one reduction from the R-UCM problem for the maximin voting rule to it. Given a R-UCM problem instance, we define a CPMW problem instance $\Gamma = (\mathcal{C}', (\succ'_i)_{i \in \mathcal{V}'}, M')$ as follows.

$$\mathcal{C}' := \mathcal{C} \cup \{w, d_1, d_2, d_3\}$$

We define \mathcal{V}' such that $d(a, b)$ is the same as the R-UCM instance, for all $a, b \in \mathcal{C}$ and $d(d_1, w) = 2k + 2, d(d_1, d_2) = 8k, d(d_2, d_3) = 8k, d(d_3, d_1) = 8k$. The existence of such a \mathcal{V}' is guaranteed from Lemma 3. Moreover, Lemma 3 also ensures that $|\mathcal{V}'|$ is $O(mc)$. The votes of the voters in M is $w \succ \dots$. Thus the *current winner* is w . The *actual winner* is defined to be z . The tie breaking rule is $\succ_t = w \succ z \succ \dots$, where z is the candidate whom the manipulators in M want to make winner in the R-UCM problem instance. Clearly this reduction takes polynomial amount of time. Now we show that, M is a coalition of possible manipulators iff z can be made a winner.

The *if* part is as follows. Let $\succ_i, i \in M$ be the votes that make z win. We can assume that z is the most preferred candidate in all the preferences $\succ_i, i \in M$. Now consider the preferences for the voters in M is follows.

$$\succ'_i := d_1 \succ d_2 \succ d_3 \succ w \succ_i, i \in M$$

The score of every candidate in \mathcal{C} is not more than z . The score of z is $-3k$. The score of w is $-3k - 2$ and the scores of d_1, d_2 , and d_3 are less than $-3k$. Hence, M is a coalition of possible manipulators with the actual preferences $\succ'_i := d_1 \succ d_2 \succ d_3 \succ w \succ_i, i \in M$.

The *only if* part is as follows. Suppose M is a coalition of possible manipulators with actual preferences $\succ'_i, i \in M$. Consider the preferences $\succ'_i, i \in M$, but restricted to the set \mathcal{C} only. Call them $\succ_i, i \in M$. We claim that $\succ_i, i \in M$ with the votes from \mathcal{V} makes z win the election. If not then, there exists a candidate, say $a \in \mathcal{C}$, whose score is strictly more than the score of z - this is so because the tie breaking rule is in favor of z . But this contradicts the fact that z wins the election when the voters in M vote $\succ'_i, i \in M$ along with the votes from \mathcal{V}' . \square

4.3 Bucklin Voting Rule

In this subsection, we design polynomial time algorithms for both the CPMW and the CPM problem for the Bucklin voting rule. Again, we begin by showing that if there are profiles witnessing manipulation, then there exist profiles that do so with some additional structure, which will subsequently be exploited by our algorithm.

Lemma 4. *Consider a preference profile $(\succ_i)_{i \in \mathcal{V}}$, where x is the winner with respect to the Bucklin voting rule. Suppose a subset of voters $M \subset \mathcal{V}$ form a coalition of possible manipulators. Let y be the actual winner. Then there exist preferences $(\succ'_i)_{i \in M}$ such that y is a Bucklin winner in $((\succ_i)_{i \in \mathcal{V} \setminus M}, (\succ'_i)_{i \in M})$, and further:*

1. y immediately follows x in each \succ'_i .
2. The rank of x in each \succ'_i is in one of the following - first, $b(y) - 1, b(y), b(y) + 1$, where $b(y)$ be the Bucklin score of y in $((\succ_i)_{i \in \mathcal{V} \setminus M}, (\succ'_i)_{i \in M})$.

Proof: From Definition 3, y 's rank must be worse than x 's rank in each \succ'_i . We now exchange the position of y with the candidate which immediately follows x in \succ'_i . This process does not decrease Bucklin score of any candidate except possibly y 's, and x 's score does not increase. Hence y will continue to win and thus \succ'_i satisfies the first condition.

Now to begin with, we assume that \succ'_i satisfies the first condition. If the position of x in \succ'_i is $b(y) - 1$ or $b(y)$, we do not change it. If x is above $b(y) - 1$ in \succ'_i , then move x and y at the first and second positions respectively. Similarly if x is below $b(y) + 1$ in \succ'_i , then move x and y at the $b(y) + 1$ and $b(y) + 2$ positions respectively. This process does not decrease score of any candidate except y because the Bucklin score of x is at least $b(y)$. The transformation cannot increase the score y since its position has only been improved. Hence y continues to win and thus \succ'_i satisfies the second condition. \square

Lemma 4 leads us to the following theorem.

Theorem 6. *The CPMW problem and the CPM problems for Bucklin voting rule are in P for any coalition of size. Therefore, by Proposition 1, the CPMSW and the CPMS problems are in P when the maximum coalition size $k = O(1)$.*

Proof: Proposition 1 says that it is enough to prove that the CPMW problem is in P. Let x be the *current winner* and y be the given *actual winner*. For any final Bucklin score $b(y)$ of y , there are polynomially many possibilities for the positions of x and y in the profile of $\succ_i, i \in M$, since Bucklin voting rule is anonymous. Once the positions of x and y is fixed, we try to fill the top $b(y)$ positions of each \succ'_i - place a candidate in an empty position above $b(y)$ in any \succ'_i if doing so does not make y lose the election. If we are able to successfully fill the top $b(y)$ positions of all \succ'_i for all $i \in M$, then M is a coalition of possible manipulators. If the above process fails for all possible above mentioned positions of x and y and all possible guesses of $b(y)$, then M is not a coalition of possible manipulators. Clearly the above algorithm runs in $poly(m, n)$ time.

The proof of correctness is as follows. If the algorithm outputs that M is a coalition of possible manipulators, then it actually has constructed \succ'_i for all $i \in M$ with respect to which they form a coalition of possible manipulators. On the other hand, if they form a coalition of possible manipulators, then Lemma 4 ensures that our algorithm explores all the sufficient positions of x and y in \succ'_i for all $i \in M$. Now if M is a possible coalition of manipulators, then the corresponding positions for x and y have also been searched. Our greedy algorithm must find it since permuting the candidates except x and z which are ranked above $b(y)$ in \succ'_i cannot stop y to win the election since the Bucklin score of other candidates except y is at least $b(y)$. \square

4.4 STV Voting Rule

Next, we prove that the CPMW and the CPM problems for STV rule is in NPC. To this end, we reduce from the Exact Cover by 3-Sets Problem (X3C), which is known to be in NPC [18]. The X3C problem is as follows.

Definition 9. (*X3C Problem*)

Given a set S of cardinality n and m subsets $S_1, S_2, \dots, S_m \subset S$ with $|S_i| = 3, \forall i = 1, \dots, m$, does

there exist an index set $I \subseteq \{1, \dots, m\}$ with $|I| = \frac{|S|}{3}$ such that $\cup_{i \in I} S_i = S$.

Theorem 7. *The CPM problem for STV rule is in NPC, for a coalition of size 1.*

Proof sketch: Clearly the problem is in NP. To show NP hardness, we show a many-one reduction from the X3C problem to it. The reduction is analogous to the reduction given in [1]. Hence, we give a proof sketch only. Given an X3C instance, we construct an election as follows. The unspecified positions can be filled in any arbitrary way. The candidate set is as follows.

$$\begin{aligned} \mathcal{C} = \{x, y\} \cup & \{a_1, \dots, a_m\} \cup \{\bar{a}_1, \dots, \bar{a}_m\} \\ & \cup \{b_1, \dots, b_m\} \cup \{\bar{b}_1, \dots, \bar{b}_m\} \\ & \cup \{d_0, \dots, d_n\} \cup \{g_1, \dots, g_m\} \end{aligned}$$

The votes are as follows.

- $12m$ votes for $y \succ x \succ \dots$
- $12m - 1$ votes for $x \succ y \succ \dots$
- $10m + \frac{2n}{3}$ votes for $d_0 \succ x \succ y \succ \dots$
- $12m - 2$ votes for $d_i \succ x \succ y \succ \dots, \forall i \in [n]$
- $12m$ votes for $g_i \succ x \succ y \succ \dots, \forall i \in [m]$
- $6m + 4i - 5$ votes for $b_i \succ \bar{b}_i \succ x \succ y \succ \dots, \forall i \in [m]$
- 2 votes for $b_i \succ d_j \succ x \succ y \succ \dots, \forall i \in [m], \forall j \in S_i$
- $6m + 4i - 1$ votes for $\bar{b}_i \succ b_i \succ x \succ y \succ \dots, \forall i \in [m]$
- 2 votes for $\bar{b}_i \succ d_0 \succ x \succ y \succ \dots, \forall i \in [m]$
- $6m + 4i - 3$ votes for $a_i \succ g_i \succ x \succ y \succ \dots, \forall i \in [m]$
- 1 vote for $a_i \succ b_i \succ g_i \succ x \succ y \succ \dots, \forall i \in [m]$
- 2 votes for $a_i \succ \bar{a}_i \succ g_i \succ x \succ y \succ \dots, \forall i \in [m]$
- $6m + 4i - 3$ votes for $\bar{a}_i \succ g_i \succ x \succ y \succ \dots, \forall i \in [m]$
- 1 vote for $\bar{a}_i \succ \bar{b}_i \succ g_i \succ x \succ y \succ \dots, \forall i \in [m]$
- 2 votes for $\bar{a}_i \succ a_i \succ g_i \succ x \succ y \succ \dots, \forall i \in [m]$

The tie breaking rule is $\succ_t = \dots \succ x$. The vote of v is $x \succ \dots$. We claim that v is a possible manipulator iff the X3C is a *yes* instance. Notice that, of the first $3m$ candidates to be eliminated, $2m$ of them are a_1, \dots, a_m and $\bar{a}_1, \dots, \bar{a}_m$. Also exactly one of b_i and \bar{b}_i will be eliminated among the first $3m$ candidates to be eliminated because if one of b_i, \bar{b}_i then the other's score exceeds $12m$. We show that the winner is either x or y irrespective of the vote of one more candidate. Let $J := \{j : b_j \text{ is eliminated before } \bar{b}_j\}$. If J is an index of set cover then the winner is y . This can be seen as follows. Consider the situation after the first $3m$ eliminations. Let $i \in S_j$ for some $j \in J$. Then b_j has been eliminated and thus the score of d_i is at least $12m$. Since J is an index of a set cover, every d_i 's score is at least $12m$. Notice that \bar{b}_j has been eliminated for all $j \notin J$. Thus the revised score of d_0 is at least $12m$. After the first $3m$ eliminations, the remaining candidates are $x, y, \{d_i : i \in [n]\}, \{g_i : i \in [m]\}, \{b_j : j \notin J\}, \{\bar{b}_j : j \in J\}$. All the remaining candidates except x has score at least $12m$ and x 's score is $12m - 1$. Hence x will be eliminated next which makes y 's score at least $24m - 1$.

Next d_i 's will get eliminated which will in turn make y 's score $(12n + 36)m - 1$. At this point g_i 's score is at most $32m$. Also all the remaining b_i and \bar{b}_i 's score is at most $32m$. Since each of the remaining candidate's scores gets transferred to y once they are eliminated, y is the winner.

Now we show that, if J is not an index of set cover then the winner is x . This can be seen as follows. If $|J| > \frac{n}{3}$, then the number of \bar{b}_j that gets eliminated in the first $3m$ iterations is less than $m - \frac{n}{3}$. This makes the score of d_0 at most $12m - 2$. Hence d_0 gets eliminated before x and all its scores gets transferred to x . This makes the elimination of x impossible before y and makes x the winner of the election.

If $|J| \leq \frac{n}{3}$ and there exists an $i \in S$ that is not covered by the corresponding set cover, then d_i gets eliminated before x with a score of $12m - 2$ and its score gets transferred to x . This makes x win the election.

Hence y can win iff X3C is a *yes* instance. Also notice that if y can win the election, then it can do so with the voter v voting a preference like $\dots \succ x \succ y \succ \dots$. \square

From the proof of the above theorem, we have the following corollary by specifying y as the *actual winner* for the CPMW problem.

Corollary 2. *The CPMW problem for STV rule is in NPC, for a coalition of size 1.*

5. CONCLUSION

In this work, we have initiated a promising research direction for detecting manipulation in elections. We have proposed the notion of *possible* manipulation and explored several concrete computational problems, which we believe to be important in the context of voting theory. These problems involve identifying if a given set of voters are possible manipulators (with or without a specified candidate winner). We have also studied the search versions of these problems, where the goal is to simply detect the presence of possible manipulation with the maximum coalition size. We believe there is theoretical as well as practical interest in studying the proposed problems. We have provided algorithms and hardness results for many common voting rules.

In this work, we considered elections with unweighted voters only. An immediate future research direction is to study the complexity of these problems in weighted elections. Further, verifying the number of false manipulators that this model catches in a real or synthetic data set, where, we already have some knowledge about the manipulators, would be interesting. It is our conviction that both the problems that we have studied here have initiated an interesting research direction with significant promise and potential for future work.

6. ACKNOWLEDGEMENT

We acknowledge the anonymous reviewers of AAMAS 2014 for providing constructive comments to improve the paper. Neeldhara Misra is supported by the INSPIRE Faculty Scheme, DST India (project DSTO-1209).

REFERENCES

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

- [2] J. Bartholdi, C. Tovey, and M. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare (SCW)*, 6(3):227–241, 1989.
- [3] N. Betzler, R. Niedermeier, and G. J. Woeginger. Unweighted coalitional manipulation under the borda rule is NP-hard. In *IJCAI*, volume 11, pages 55–60, 2011.
- [4] V. Conitzer and T. Sandholm. Nonexistence of voting rules that are usually hard to manipulate. In *International Conference on Artificial Intelligence (AAAI)*, volume 6, pages 627–634, 2006.
- [5] V. Conitzer, T. Sandholm, and J. Lang. When are elections with few candidates hard to manipulate? *Journal of the ACM (JACM)*, 54(3):14, 2007.
- [6] J. Davies, G. Katsirelos, N. Narodytska, and T. Walsh. Complexity of and algorithms for borda manipulation. In *Proceedings of the International Conference on Artificial Intelligence (AAAI)*, pages 657–662, 2011.
- [7] P. Dey and Y. Narahari. Asymptotic collusion-proofness of voting rules: the case of large number of candidates. In *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1419–1420. International Foundation for Autonomous Agents and Multiagent Systems, 2014.
- [8] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proceedings of the 10th International Conference on World Wide Web (WWW)*, pages 613–622. ACM, 2001.
- [9] E. Elkind and G. Erdélyi. Manipulation under voting rule uncertainty. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 627–634. International Foundation for Autonomous Agents and Multiagent Systems, 2012.
- [10] E. Elkind and H. Lipmaa. Hybrid voting protocols and hardness of manipulation. In *Algorithms and Computation*, pages 206–215. Springer, 2005.
- [11] P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra. How hard is bribery in elections? *Journal of Artificial Intelligence Research (JAIR)*, 35(2):485, 2009.
- [12] P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra. Using complexity to protect elections. *Communications of the ACM*, 53(11):74–82, 2010.
- [13] P. Faliszewski, E. Hemaspaandra, and L. A. Hemaspaandra. Weighted electoral control. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 367–374. International Foundation for Autonomous Agents and Multiagent Systems, 2013.
- [14] P. Faliszewski and A. D. Procaccia. Ai’s war on manipulation: Are we winning? *AI Magazine*, 31(4):53–64, 2010.
- [15] P. Faliszewski, Y. Reisch, J. Rothe, and L. Schend. Complexity of manipulation, bribery, and campaign management in bucklin and fallback voting. In *Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1357–1358. International Foundation for Autonomous Agents and Multiagent Systems, 2014.
- [16] E. Friedgut, G. Kalai, and N. Nisan. Elections can be manipulated often. In *IEEE 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 243–249. IEEE, 2008.
- [17] W. Gaertner. *Domain Conditions in Social Choice Theory*. Cambridge University Press, 2001.
- [18] M. R. Garey and D. S. Johnson. *Computers and Intractability*, volume 174. freeman New York, 1979.
- [19] S. Gaspers, T. Kalinowski, N. Narodytska, and T. Walsh. Coalitional manipulation for schulze’s rule. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 431–438. International Foundation for Autonomous Agents and Multiagent Systems, 2013.
- [20] A. Gibbard. Manipulation of voting schemes: a general result. *Econometrica: Journal of the Econometric Society*, pages 587–601, 1973.
- [21] M. Isaksson, G. Kindler, and E. Mossel. The geometry of manipulation - a quantitative proof of the gibbard-satterthwaite theorem. *Combinatorica*, 32(2):221–250, 2012.
- [22] A. Mas-Collel, M. D. Whinston, and J. Green. *Microeconomic theory*, 1995.
- [23] D. C. McGarvey. A theorem on the construction of voting paradoxes. *Econometrica: Journal of the Econometric Society*, pages 608–610, 1953.
- [24] N. Narodytska and T. Walsh. Manipulating two stage voting rules. In *Proceedings of the 12th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 423–430. International Foundation for Autonomous Agents and Multiagent Systems, 2013.
- [25] N. Narodytska, T. Walsh, and L. Xia. Manipulation of nanson’s and baldwin’s rules. In *Proceedings of the International Conference on Artificial Intelligence (AAAI)*, pages 713–718, 2011.
- [26] S. Obraztsova and E. Elkind. Optimal manipulation of voting rules. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 619–626. International Foundation for Autonomous Agents and Multiagent Systems, 2012.
- [27] S. Obraztsova, E. Elkind, and N. Hazon. Ties matter: Complexity of voting manipulation revisited. In *The 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 71–78. International Foundation for Autonomous Agents and Multiagent Systems, 2011.
- [28] D. M. Pennock, E. Horvitz, and C. L. Giles. Social choice theory and recommender systems: Analysis of the axiomatic foundations of collaborative filtering. In *Proceedings of the 17th International Conference on Artificial Intelligence (AAAI)*, 2000.
- [29] A. Procaccia and J. Rosenschein. Junta distributions and the average-case complexity of manipulating elections. In *Proceedings of the Fifth International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 497–504. ACM, 2006.

- [30] A. Procaccia and J. Rosenschein. Average-case tractability of manipulation in voting via the fraction of manipulators. In *Proceedings of the 6th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, volume 7, 2007.
- [31] M. Satterthwaite. Strategy-proofness and arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory (JET)*, 10(2):187–217, 1975.
- [32] T. Walsh. An empirical study of the manipulability of single transferable voting. In *Proceedings of the 19th European Conference on Artificial Intelligence (ECAI)*, pages 257–262, 2010.
- [33] T. Walsh. Is computational complexity a barrier to manipulation? *Annals of Mathematics and Artificial Intelligence*, 62(1-2):7–26, 2011.
- [34] L. Xia. Computing the margin of victory for various voting rules. In *Proceedings of the 13th ACM Conference on Electronic Commerce (EC)*, pages 982–999. ACM, 2012.
- [35] L. Xia and V. Conitzer. Determining possible and necessary winners under common voting rules given partial orders. In *International Conference on Artificial Intelligence (AAAI)*, volume 8, pages 196–201, 2008.
- [36] L. Xia and V. Conitzer. Generalized scoring rules and the frequency of coalitional manipulability. In *Proceedings of the 9th ACM conference on Electronic Commerce (EC)*, pages 109–118. ACM, 2008.
- [37] L. Xia and V. Conitzer. A sufficient condition for voting rules to be frequently manipulable. In *Proceedings of the 9th ACM conference on Electronic Commerce (EC)*, pages 99–108. ACM, 2008.
- [38] L. Xia, V. Conitzer, and A. D. Procaccia. A scheduling approach to coalitional manipulation. In *Proceedings of the 11th ACM conference on Electronic Commerce (EC)*, pages 275–284. ACM, 2010.
- [39] L. Xia, M. Zuckerman, A. D. Procaccia, V. Conitzer, and J. S. Rosenschein. Complexity of unweighted coalitional manipulation under some common voting rules. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, volume 9, pages 348–353, 2009.
- [40] M. Zuckerman, O. Lev, and J. S. Rosenschein. An algorithm for the coalitional manipulation problem under maximin. In *The 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 845–852. International Foundation for Autonomous Agents and Multiagent Systems, 2011.