Manipulation Under Voting Rule Uncertainty

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

Gábor Erdélyi
University of Siegen
Germany
erdelyi@wiwi.uni-siegen.de

ABSTRACT

An important research topic in the field of computational social choice is the complexity of various forms of dishonest behavior, such as manipulation, control, and bribery. While much of the work on this topic assumes that the cheating party has full information about the election, recently there have been a number of attempts to gauge the complexity of non-truthful behavior under uncertainty about the voters' preferences. In this paper, we analyze the complexity of (coalitional) manipulation for the setting where there is uncertainty about the voting rule: the manipulator(s) know that the election will be conducted using a voting rule from a given list, and need to select their votes so as to succeed no matter which voting rule will eventually be chosen. We identify a large class of voting rules such that arbitrary combinations of rules from this class are easy to manipulate; in particular, we show that this is the case for single-voter manipulation and essentially all easy-to-manipulate voting rules, and for coalitional manipulation and k-approval. While a combination of a hard-to-manipulate rule with an easy-to-manipulate one is usually hard to manipulate—we prove this in the context of coalitional manipulation for several combinations of prominent voting rules—we also provide counterexamples showing that this is not always the case.

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

Economics, Theory

Keywords

Computational Social Choice, Manipulation, Uncertainty

1. INTRODUCTION

Voting is an established framework for making collective decisions, and as such has applications in settings that range from political elections to faculty hiring decisions, selecting the winners of singing competitions, and the design of multiagent systems. In some of these settings, the number of candidates and/or voters can be large, yet the decision needs to be made quickly. Whenever this is the case, the algorithmic complexity of, on the one hand, winner determination and, on the other hand, various forms of dishonest behavior in elections, plays an important role in the selection of a voting rule: we want the former to be as low as possible, while keeping the latter as high as possible.

Traditionally, the complexity of voting rules is studied under the full information assumption: for instance, in the single-voter manipulation problem, which is perhaps one of the most fundamental problems in the complexity-theoretic analysis of voting rules, it is assumed that the manipulator knows the set of candidates, the number and the true preferences of all honest voters, and, crucially, the voting rule. However, it is widely recognized that this assumption is not always realistic, and recently a number of papers tried to analyze the complexity of cheating in elections and/or determining the likely election winners under various forms of uncertainty about the election (see Section 1.1 for an overview).

In this paper, we study the complexity of manipulation (both by a single voter and by a coalition of voters) in settings where there is uncertainty about the voting rule itself. That is, we assume that the manipulator(s) know that the voting rule belongs to a certain (finite or infinite) family of rules $\widehat{\mathcal{F}}$, and they want to select their votes so as to ensure that their preferred candidate wins, no matter which of the rules in $\widehat{\mathcal{F}}$ is chosen.

Admittedly, in political elections the voting rule to be used is typically known before the votes are cast, and the manipulator would be well advised to fully understand the voting rule before modifying her vote. However, in other applications of voting this is not always the case. For instance, it is not unusual for a university department to ask graduate students to provide a ranking of faculty candidates; however, the graduate students are not told how the hiring committee makes its decision (anecdotally, a wide variety of voting rules can be used for this purpose). Another example is provided by conference reviewing: at some point in the decision-making process, the program committee members may be asked to rank the papers whose fate has not been decided yet; the PC chair will then aggregate the rankings in a way that has not been announced to the PC members (and may, in fact, be unknown to the PC chair when she initiates the process). In some of these settings, the voters may believe that the voting rule will be chosen from a specific family of rules: for instance, the voters may know that the rule to be used is a scoring rule, or, more narrowly, a k-Approval rule (with the value of k unknown), or a Condorcet-consistent rule (see Section 2 for definitions); the situation where the voters know the voting correspondence, but not the tie-breaking rule is also captured by this description. They may then want to select their votes so that their favorite candidate wins the election *no matter which of the voting rules in this family is chosen*.

We study the complexity of this problem for several families of

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

voting rules. We limit ourselves to the setting of voting manipulation (either by a single voter or by a coalition of voters), though one can ask the same question in the context of election control or bribery (see, e.g., [13] for the definitions and a survey of recent results for these problems). We mostly focus on families that consist of a small number (usually, two) prominent voting rules, such as Plurality, k-Approval, Borda, Copeland, Maximin and STV. Our goal is not to classify all such combinations or rules: rather, we try to illustrate the general techniques that can be used for the analysis of such settings.

One would expect a combination of easy-to-manipulate rules to be easy to manipulate, and a combination of several hard-to-manipulate rules or an easy-to-manipulate one with a hard-to-manipulate one to be hard to manipulate. Our results for classic voting rules mostly confirm this intuition, with the exception of settings where we combine a hard-to-manipulate rule with one that is very indecisive. However, we show that these results are not universal: we provide an example of two hard-to-manipulate rules whose combination is easy to manipulate, as well as an example of two easy-to-manipulate rules whose combination is hard to manipulate. While the rules used in these constructions are fairly artificial, they nevertheless illustrate interesting aspects of our problem.

1.1 Related Work

Our works fits into the stream of research on winner determination and voting manipulation under uncertainty. In the context of winner determination, perhaps the most prominent problem in this category is the possible/necessary winner problem [16], where the voting rule is public information, but, for each voter, only a partial order over the candidates in known; the goal is to determine if a candidate wins the election for *some* way (the *possible winner*) or for *every* way (the *necessary winner*) of completing the voters' preferences; a probabilistic variant of this problem has also been considered [1]. Our problem is more similar in flavor to the necessary winner problem, as the manipulator has to succeed for *all* voting rules in the family.

Uncertainty about the voting rule has been recently investigated by Baumeister et al. [5], who also consider the situation where the voting rule will be chosen from a fixed set. In contrast to our work, they assume that all voters' preferences are known, and ask if there is a voting rule that makes a certain candidate a winner with respect to these preferences; thus, in their work the manipulating party is the election authority rather than one of the voters.

Our problem is, in a sense, dual to the one considered by Conitzer et al. [7]: in their model the voting rule is known, but the preferences of some of the honest voters are (partially) unknown; they ask if the manipulator can cast a vote that improves the outcome (from his perspective) for *every* realization of the honest voters' preferences; thus, just like us, they assume an adversarial environment.

There has also been some work on settings where the *effects of the manipulator's actions* are uncertain. This is the case, for instance, for the model of safe strategic voting [19], where one voter announces a manipulative vote, and one or more voters with the same true preferences may follow suit; the original manipulator does not know how many followers he will have and needs to choose the vote so as to improve the outcome for *some* number of followers, while ensuring that the outcome does not get worse for *any* number of followers. Another example is cloning [9], where the cheating party clones one or more candidates; the voters are assumed to rank the clones of a given candidate consecutively, but the exact order of the clones in voters' preferences is unknown. Our work is most similar to the variant of this problem known as 1-CLONING, where the cheating party has to succeed *no matter how the voters order the clones*.

Finally, we remark that the idea of combining two or more voting rules has been considered in early work on computational social choice [10, 14]; however, in both of these papers, voting rules are combined in a way that is very different from our work.

2. PRELIMINARIES

Given a finite set S, we denote by $\mathcal{L}(S)$ the space of all linear orders over S. An *election* is a triple $E = (C, V, \mathcal{R})$, where $C = \{c_1, \dots, c_m\}$ is the set of *candidates*, V is the set of *voters*, $|V| = n$, and $\mathcal{R} = (R_1, \dots, R_n)$ is the *preference profile*, i.e., a collection of linear orders over C. The order R_i is called the *preference order*, or *vote*, of voter i; we will also denote R_i by $\succ_i$. When $a \succ_i b$ for some $a, b \in C$, we say that voter i *prefers* a to b. A candidate a is said to be the *top-ranked* candidate of voter i, or *receive a first-place vote* from i, if $a \succ_i b$ for all $b \in C \setminus \{i\}$.

A *voting correspondence* $\mathcal{F}$ is a mapping that, given an election $E = (C, V, \mathcal{R})$ outputs a non-empty subset $S \subseteq C$; we write $S = \mathcal{F}(E)$. The elements of the set S are called the *winners* of the election E under $\mathcal{F}$. If $|\mathcal{F}(E)| = 1$ for any election E, the mapping $\mathcal{F}$ is called a *voting rule*; whenever this is the case, we abuse notation and write $\mathcal{F}(\mathcal{R}) = c$ instead of $\mathcal{F}(\mathcal{R}) = \{c\}$. We will sometimes abuse terminology and refer to voting correspondences as voting rules.

A voting correspondence $\mathcal{F}$ is said to be *neutral* if renaming the candidates does not alter the set of winners: that is, for any election $E = (C, V, \mathcal{R})$ and any permutation π of the set C, the election E' obtained by replacing each candidate c in $\mathcal{R}$ by $\pi(c)$ satisfies $\mathcal{F}(E') = \{\pi(c) \mid c \in \mathcal{F}(E)\}$. $\mathcal{F}$ is said to be *monotone* if promoting a winning candidate does not make him lose the election: if $c \in \mathcal{F}(E)$, then $c \in \mathcal{F}(E')$, where E' is obtained from E by swapping c with the candidate ranked just above c in some vote (this notion of monotonicity is sometimes referred to as *weak* monotonicity).

Voting rules We will now describe the voting rules (correspondences) considered in this paper. For all rules that assign scores to candidates (i.e., scoring rules, Copeland, and Maximin), the winners are the candidates with the highest scores.

Scoring rules Any vector $\alpha = (\alpha_1, \dots, \alpha_m) \in \mathbb{R}^m$ such that $\alpha_1 \geq \dots \geq \alpha_m$ defines a *scoring rule* $\mathcal{F}_\alpha$ over a set of candidates of size m: a candidate receives α_j points from each voter who ranks him in the j-th position, and the score of a candidate is the total number of points he receives from all voters. The vector α is called a *scoring vector*. We assume without loss of generality that the entries of α are nonnegative integers given in binary. As we require voting rules to be defined for any number of candidates, we will consider *families* of scoring rules: one for every possible number of candidates. We denote such families by $\{\mathcal{F}_{\alpha^m}\}_{m=1,\dots}$, where $\alpha^m = (\alpha^m_1, \dots, \alpha^m_m)$ is the scoring vector of length m. Two well-known examples of such families are Borda, given by $\alpha^m = (m-1, \dots, 1, 0)$ for all $m > 1$, and k-Approval, given by $\alpha^m_i = 1$ if $i \leq k$, $\alpha^m_i = 0$ if $i > k$. The 1-Approval rule is also known as Plurality.

Condorcet We say that a candidate a wins a *pairwise election* against b if more than half of the voters prefer a to b; if exactly half of the voters prefer a to b, then a is said to *tie* his pairwise election against b. A candidate is said to be a *Condorcet winner* if he wins pairwise elections against all other candidates. The Condorcet rule outputs the Condorcet winner if it exists; otherwise, it outputs the set of all candidates (recall that a voting correspondence should always output a non-empty set of winners).

Copeland Given a rational value $\alpha \in [0, 1]$, under the $\mathsf{Copeland}^\alpha$ rule each candidate gets 1 point for each pairwise election he wins and α points for each pairwise election he ties.

Maximin The Maximin score of a candidate $c \in C$ is equal to the number of votes he gets in his worst pairwise election, i.e., $\min_{d \in C \setminus \{c\}} |\{i \mid c \succ_i d\}|$.

STV Under the STV rule, the election proceeds in rounds. During each round, the candidate with the lowest Plurality score is eliminated, and the candidates' Plurality scores are recomputed. The winner is the candidate that survives till the end. If several candidates have the lowest Plurality score (we will refer to this situation as an *intermediate tie*), we assume that the candidate to be eliminated is chosen according to the lexicographic order over the candidates: if S is the set of candidates that have the lowest Plurality score in some round, we eliminate the candidate c_j such that $j \geq i$ for all $c_i \in S$. We remark that STV, as defined here, always has a single winner; however, because of the lexicographic tie-breaking rule it is not neutral.

3. PROBLEM STATEMENT

We assume that we are given a collection $\widehat{\mathcal{F}} = \{\mathcal{F}_i\}_{i \in I}$ of voting correspondences. The set $\widehat{\mathcal{F}}$ can be finite of infinite; for instance, $\widehat{\mathcal{F}}$ can be the set of all (families of) scoring rules, in which case it is infinite. When $\widehat{\mathcal{F}}$ is infinite, we assume that it admits a succinct description; if $\widehat{\mathcal{F}}$ is finite, it is assumed to be listed explicitly.

We consider the complexity of (coalitional) manipulation in elections when the manipulator does not know which of the voting rules in $\widehat{\mathcal{F}}$ will be selected. We state our definitions in the *unique winner* model, i.e., we assume that the manipulator's goal is to make its preferred candidate the unique winner with respect to each of the voting correspondences in $\widehat{\mathcal{F}}$; however, most of our results remain true in the *co-winner* model, where the manipulator would like to ensure that its preferred candidate is one of the winners under each of the voting correspondences in $\widehat{\mathcal{F}}$.

Name: $\widehat{\mathcal{F}}$-MANIPULATION BY SINGLE VOTER (SM)

Input: An election (C, V) with $|C| = m$, $|V| = n - 1$, a preference profile $\mathcal{R} = (R_1, \dots, R_{n-1})$, and a candidate $p \in C$.

Question: Is there a vote $L \in \mathcal{L}(C)$ such that p is the unique winner in $(\mathcal{R}, L)$ with respect to each of the voting correspondences in $\widehat{\mathcal{F}}$?

Voters $1, \dots, n - 1$ are referred to as the *honest* voters, and the last voter (the one who submits vote L and wants p to win) is referred to as the *manipulator*.

Name: $\widehat{\mathcal{F}}$-COALITIONAL MANIPULATION (CM)

Input: An election (C, V) with $|C| = m$, $|V| = h$, a set M, $|M| = s = n - h$, a preference profile $\mathcal{R} = (R_1, \dots, R_h)$, and a candidate $p \in C$.

Question: Is there a profile $\mathcal{L} = (L_1, \dots, L_s) \in \mathcal{L}^s(C)$ such that p is the unique winner in $(\mathcal{R}, \mathcal{L})$ with respect to each of the voting correspondences in $\widehat{\mathcal{F}}$?

If $\widehat{\mathcal{F}}$ is finite, we say that an algorithm $\mathcal{A}$ for $\widehat{\mathcal{F}}$-SM or $\widehat{\mathcal{F}}$-CM is a polynomial-time algorithm if its running time is polynomial in n, m, and $|\widehat{\mathcal{F}}|$; if $\widehat{\mathcal{F}}$ is infinite, we require the running time of $\mathcal{A}$ to be polynomial in n and m. We remark that $\widehat{\mathcal{F}}$-SM (respectively, $\widehat{\mathcal{F}}$-CM) is in NP for any finite collection $\widehat{\mathcal{F}}$ of polynomially computable voting rules: it suffices to guess a manipulative vote L (respectively, a list $(L_1, \dots, L_s)$ of manipulative votes) and verify that it makes p the unique winner under every rule in $\widehat{\mathcal{F}}$. Thus, in what follows, when proving that these problems are NP-complete for some finite $\widehat{\mathcal{F}}$, we will only provide an NP-hardness proof.

Traditionally, the problems $\widehat{\mathcal{F}}$-SM and $\widehat{\mathcal{F}}$-CM are studied for the case $|\widehat{\mathcal{F}}| = 1$. In what follows, whenever $\widehat{\mathcal{F}} = \{\mathcal{F}\}$, we omit the curly braces and write $\mathcal{F}$-SM/CM instead of $\{\mathcal{F}\}$-SM/CM to conform with the standard notation.

4. MANIPULATION

We start by considering the SM problem. In their classic paper [3], Bartholdi, Tovey and Trick show that this problem is polynomial-time solvable for $\mathsf{Copeland}^\alpha$ (for every rational $\alpha \in [0, 1]$), Maximin, and all scoring rules (while Bartholdi et al. do not explicitly consider scoring rules other than Plurality and Borda, it is not hard to see that their algorithm works for any scoring rule).

Remarkably, for all these rules the manipulative vote can be found by essentially the same algorithm. This algorithm starts by ranking p first; it is safe to do so, because all of these rules are monotone. Note that at this point we can already compute p's final score; let us denote it by $s(p)$. The algorithm then fills up positions $2, \dots, m$ in the vote one by one. When considering position i, $i \geq 2$, it tries to place each of the still unranked candidates into this position. At this point, the identities of the candidates in positions $1, \dots, i - 1$ are already known, so one can determine the score of each candidate c if it were to be placed in position i (this is true for Copeland, Maximin and all scoring rules, but need not be true in general, even for monotone rules); let us denote this quantity by $s_i(c)$. If there exists a candidate c such that $s_i(c) < s(p)$, it is placed in position i; if there are several such candidates, one of them is selected arbitrarily. If no such candidate can be found, the algorithm reports that no manipulative vote exists.

Bartholdi et al. prove the correctness of this algorithm for all voting correspondences that (1) are monotone and (2) have the property that the score of a candidate c can be determined if we know which candidates are ranked above and below c in each vote, and the winners are the candidates with the highest score. $\mathsf{Copeland}^\alpha$, $\alpha \in \mathbb{Q} \cap [0, 1]$, Maximin, and all scoring rules satisfy both of these conditions, and STV satisfies neither of them; indeed, STV-SM is known to be NP-complete [2].

We will now show that the algorithm of Bartholdi et al. extends to $\widehat{\mathcal{F}}$-SM for any finite set $\widehat{\mathcal{F}}$ that consists of voting correspondences that satisfy (1) and (2).

THEOREM 4.1. *Let $\widehat{\mathcal{F}}$ be a finite set of voting rules such that every rule $\mathcal{F}_i \in \widehat{\mathcal{F}}$ satisfies conditions (1) and (2). Then $\widehat{\mathcal{F}}$-SM can be solved in polynomial time.*

PROOF. Let $\widehat{\mathcal{F}} = \{\mathcal{F}_1, \dots, \mathcal{F}_\ell\}$. Our algorithm proceeds in rounds: in round i, $i = 1, \dots, m$, we consider position i.

In the first round, we place p in the top position; let $s^j(p)$ denote p's score with respect to the rule $\mathcal{F}_j$, $j = 1, \dots, \ell$, in the resulting election.

Now, consider round i, $i = 2, \dots, m$. For each candidate c that has not been ranked in round $1, \dots, i - 1$, let $s_i^j(c)$ be his score under rule $\mathcal{F}^j$ if he were to be ranked in position i at this point. If there exists an unranked candidate c such that $s_i^j(c) < s^j(p)$ for all $j = 1, \dots, \ell$, we place c in position i; if there are several such candidates, we choose one of them arbitrarily. If no such candidate exists, we report that the input instance of $\widehat{\mathcal{F}}$-SM cannot be manipulated. If we manage to successfully rank all candidates, we report that there exists a successful manipulative vote; in fact, our algorithm constructs it.

Clearly, if our algorithm reports that a manipulative vote exists, this is indeed the case. Conversely, suppose that our algorithm reports that the given election cannot be manipulated. This means that during some round i, we have $s^j_i(c) \geq s^j(p)$ for some voting rule $\mathcal{F}_j \in \widehat{\mathcal{F}}$ and every candidate c that has not been ranked in rounds $1, \dots, i-1$. But then consider the execution of the original algorithm of Bartholdi et al. [3] on the instance of $\mathcal{F}_j$-SM given by the same election. The algorithm of Bartholdi et al. could have made exactly the same choices as our algorithm in rounds $1, \dots, j-1$. Therefore, it, too, would have reported that its input instance is a "no"-instance. Since the algorithm of Bartholdi et al. is known to be correct, this means that no manipulative vote could have made p the unique winner with respect to $\mathcal{F}_j$. Hence, our instance of $\widehat{\mathcal{F}}$-SM is a "no"-instance as well, which means that our algorithm is correct. $\square$

The proof of Theorem 4.1 is very simple. However, the result itself plays a key role in our understanding of single-voter manipulation under voting rule uncertainty. Indeed, to the best of our knowledge, for all classic voting rules for which single-voter manipulation is known to be easy, a manipulative vote can be constructed using the algorithm of [3]. Therefore, we cannot hope to put together two or more classic easy-to-manipulate rules so that the manipulation problem with respect to the combination of these rules is computationally hard.

One can nevertheless ask if such a combination of rules exists. We will now show that the answer to this question is "yes": we present two easy-to-manipulate rules, which we will call STV_1 and STV_2, such that STV_i-SM is polynomial-time solvable for $i = 1, 2$ but $\{\mathsf{STV}_1, \mathsf{STV}_2\}$-SM is NP-hard. Admittedly, these rules are not particularly natural; but then Theorem 4.1 shows that we cannot hope to prove a result of this type for natural voting rules.

The main idea of the construction is that each of these rules can be manipulated either by making p the STV winner or by using an easy-to-compute "trapdoor"; however, the "trapdoors" for STV_1 and STV_2 are incompatible with each other, so, to manipulate both, one needs to manipulate STV.

Formally, STV_1 is defined as follows. For $m \leq 3$, all candidates are declared to be the winners. For $m > 3$, the rule is not neutral in a very essential way: candidates c_{m-2}, c_{m-1} and c_m play a special role. Specifically, if some voter ranks c_{m-3+j} in position $m-3+j$ for $j = 1, 2, 3$, then the candidate ranked first by this voter is declared to be the election winner; if there are several such voters, the set of winners consists of these voters' top choices. Otherwise, the winner is the winner under the STV rule.

STV_2 coincides with STV_1 for $m \leq 3$. For $m > 3$, if some voter ranks c_{m-3+j} in position c_{m+1-j} for $j = 1, 2, 3$, then the candidate ranked first by this voter is declared to be the election winner (again, the election may have multiple winners if there are several such voters), and otherwise the winner is the STV winner.

THEOREM 4.2. *STV_1-SM and STV_2-SM are in* P. *However,* $\{\mathsf{STV}_1, \mathsf{STV}_2\}$-SM *is* NP-*complete.*

PROOF. Consider an instance of STV_1. Suppose that some of the honest voters rank c_{m-3+j} in position $m-3+j$ for $j = 1, 2, 3$, and let S be the set of these voters' top choices. If $S \neq \{p\}$, no matter what the manipulator does, all candidates in S will be declared the election winners, so the manipulator cannot make p the unique winner. If $S = \{p\}$, or if none of the honest voters ranks c_{m-3+j} in position $m-3+j$ for $j = 1, 2, 3$, the manipulator can rank p first and place c_{m-3+j} in position $m-3+j$ for $j = 1, 2, 3$; this would make p the unique winner. In any case, the manipulator's problem is in P. A similar argument shown that STV_2-SM is in P.

To show that $\{\mathsf{STV}_1, \mathsf{STV}_2\}$-SM is NP-hard, we will provide an NP-hardness reduction from STV-SM, which is known to be NP-complete [2].

Given an instance of STV-SM with a set of candidates $C = \{c_1, \dots, c_{m'}\}$, a set of voters V, $|V| = n-1$, a preference profile $\mathcal{R} = (R_1, \dots, R_{n-1})$ over C, and a preferred candidate $p \in C$, we will modify it as follows. We let $m = m' + 3$ and set $C' = C \cup \{c_{m-2}, c_{m-1}, c_m\}$. We ask each of the voters to rank each of the candidates in C in the same position as before, and rank c_{m-1} in position $m-2$, followed by c_{m-2} and c_m; denote the resulting preference profile by $\mathcal{R}'$.

Observe that the manipulator can make p the unique winner of this election under STV_1 either by ranking c_{m-3+j} in position $m-3+j$ for $j = 1, 2, 3$, or by making p the unique STV winner. Similarly, the manipulator can make p the unique winner of the new election under STV_2 either by ranking c_{m-3+j} in position $m+1-j$ for $j = 1, 2, 3$, or by making p the unique STV winner.

Now, suppose that the original instance of STV-SM is a "yes"-instance, and let $L \in \mathcal{L}(C)$ be the manipulative vote that makes p the STV winner in that election. Consider the vote L' obtained from L by ranking c_{m-1}, c_{m-2}, and c_m after all candidates in C (in this order). In $(\mathcal{R}', L')$, no voter ranks c_{m-2}, c_{m-1}, c_m according to either of the "trapdoors", so both in STV_1 and in STV_2 the STV rule is applied. Further, in $(\mathcal{R}', L')$ candidates c_{m-2}, c_{m-1}, c_m receive no first-place votes, so under STV they are eliminated before any candidates in C. STV then proceeds in the same way as on $(\mathcal{R}, L)$, thus making p the winner.

Conversely, suppose that there exists a vote $L' \in \mathcal{L}(C')$ such that p is the unique winner in $(\mathcal{R}', L')$ with respect to both STV_1 and STV_2. Since L cannot rank c_m in positions $m-2$ and m simultaneously, it follows that p is the STV winner in $(\mathcal{R}', L')$. Now, consider the execution of STV_1 on $(\mathcal{R}', L')$. If L' does not rank any of the candidates in $C' \setminus C$ in the top position, after the first three steps the execution of STV_1 on $(\mathcal{R}', L')$ coincides with the execution of STV on $(\mathcal{R}, L)$, where L is obtained from L' by removing c_{m-2}, c_{m-1} and c_m. Thus, in this case L is a successful manipulative vote that witnesses that the original instance of STV-SM is a "yes"-instance.

Now, suppose that L' ranks a candidate from $C' \setminus C$ first; assume without loss of generality that the top candidate in L is c_m. Then simply removing c_{m-2}, c_{m-1} and c_m from L' would not necessarily work: if the top candidate in the resulting vote receives no first-place votes in $\mathcal{R}$, this candidate would have been eliminated in the very beginning in $(\mathcal{R}', L')$, but may survive much longer in the modified election. Thus, we need a slightly different strategy. Let C_0 be the set of candidates that receive no first-place votes in $\mathcal{R}$. We construct L from L' by removing c_{m-2}, c_{m-1} and c_m and moving candidates in C_0 to the bottom of the vote (without changing the relative ordering of all other candidates). Then on $(\mathcal{R}', L')$ STV starts by eliminating c_{m-1}, c_{m-2} and the candidates in C_0. At this point, each candidate has at least one first-place vote; hence, because of our intermediate tie-breaking rule, c_m is the first candidate to be eliminated, and we are left with an election E'' over $C \setminus C_0$. On the other hand, in $(\mathcal{R}, L)$ the set of candidates with no first-place votes coincides with C_0, so after the first $|C_0|$ elimination rounds we obtain an election over $C \setminus C_0$ that coincides with E''. Hence, p is the unique STV winner in $(\mathcal{R}, L)$, and hence our original instance of STV-SM is a "yes"-instance. $\square$

We remark that Theorem 4.2 holds for coalitional manipulation as well: for the easiness result, note that the manipulators may use trapdoors to manipulate STV_1 or STV_2, and the hardness result generalizes trivially.

The next question that we would like to explore is whether a combination of an easy-to-manipulate rule with a hard-to-manipulate one is hard to manipulate. We will now illustrate that this is the case for two classic voting rules, namely, STV and Borda.

THEOREM 4.3. $\{\mathsf{Borda}, \mathsf{STV}\}$-SM *is* NP*-complete.*

PROOF. We will provide a reduction from STV-SM. Consider an instance of STV-SM given by an election (C, V) with $C = \{c_1, \dots, c_m\}$, $|V| = n-1$, a preference profile $\mathcal{R} = (R_1, \dots, R_{n-1})$, and a candidate $p \in C$; assume without loss of generality that $n \geq 3$. Suppose that p is not ranked first by any of the voters in V. Then if the manipulator does not rank p first, p get eliminated before any candidate that has a positive Plurality score in (C, V) and therefore does not win the election. Hence, the manipulator has to rank p first. Observe also that the rest of the manipulator's vote does not matter in this case: it can only impact the candidate elimination process after p is eliminated, at which point p has already lost the election. Thus, if no voter in V ranks p first, the manipulator's problem is in P: the manipulator should rank p first and check if this achieves the desired result. We can therefore assume without loss of generality that in our input instance of STV-SM candidate p receives at least one first-place vote.

Thus, assume that p is the top candidate of voter 1. Let $D = \{c_{im+j} \mid i = 1, \dots, n, j = 1, \dots, m\}$, and set $C' = C \cup D$. Modify all votes in $\mathcal{R}$ by inserting the candidates in D right below p in each vote, in an arbitrary order; let $\mathcal{R}'$ be the resulting profile.

Let $s(c)$ denote the Borda score of a candidate $c \in C$ in $(C, V, \mathcal{R})$, and let $s'(c)$ denote his score in $(C', V, \mathcal{R}')$. We have $s(c) \leq (n-1)(m-1)$ for all $c \in C$. Moreover, we have $s'(p) = s(p) + mn(n-1)$, as p gets mn extra points from each vote. On the other hand, every other candidate in C gets at most $mn(n-2)$ extra points from voters $2, \dots, n-1$ and no extra points from voter 1. Thus, for any $c \in C \setminus \{p\}$ we have

$$s'(c) \leq s(c) + mn(n-2) \leq mn(n-1) - m - n + 1 < s'(p) - m.$$

Also, the Borda score of any $d \in D$ in $(C', V, \mathcal{R}')$ is less than $s'(p)$. Thus, if the manipulator ranks the candidates in C in top m positions, p is the unique Borda winner of the resulting election.

On the other hand, no matter how the manipulator votes, under STV all candidates in D will be eliminated before all candidates in C that have a non-zero Plurality score: indeed, the Plurality score of each $d \in D$ is at most 1, and the intermediate tie-breaking rule favors candidates in C over those in D.

We are now ready to show that our reduction is correct. Let L be a successful manipulative vote for the original instance, and let C_0 be the set of all candidates in C with no first-place votes in $(\mathcal{R}, L)$. Note that the candidates in C_0 are eliminated in the first $|C_0|$ rounds of STV. Now, consider the vote L' obtained from L by ranking the candidates in D in positions $m+1, \dots, m(n+1)$. In the election $(\mathcal{R}', L')$ candidate p has the highest Borda score. Moreover, under STV we will first eliminate all candidates in $C_0 \cup D$. At this point, we obtain the same election as after $|C_0|$ rounds of STV on $(\mathcal{R}, L)$—and hence the same winner. Thus, L' is a successful manipulative vote in the new election.

Conversely, suppose that $L' \in \mathcal{L}(C \cup D)$ is such that in $(\mathcal{R}', L')$ candidate p is both the unique Borda winner and the (unique) STV winner. Let C'_0 be the set of candidates in C that have no first-place votes in $(\mathcal{R}', L')$. When we execute STV on $(\mathcal{R}', L')$, we eliminate all candidates in $D \cup C'_0$ prior to eliminating any of the candidates in $C \setminus C'_0$. Let L be the vote in $\mathcal{L}(C)$ obtained by deleting all candidates in D from L' and moving all candidates in C'_0 to the bottom $|C'_0|$ positions (without changing the relative ordering of the candidates in $C \setminus C'_0$). Then C'_0 is exactly the set of candidates in C who have no first-place votes in $(\mathcal{R}, L)$. Therefore, when we execute STV on $(\mathcal{R}, L)$, we eliminate all candidates in C'_0 prior to eliminating any candidates in $C \setminus C'_0$. Thus, the profile obtained after running STV for $|D| + |C'_0|$ steps on $(\mathcal{R}', L')$ coincides with the profile obtained after running STV for $|C'_0|$ steps on $(\mathcal{R}, L)$. Thus, L is a successful manipulative vote for the original election. $\square$

Another interesting (and arguably natural) combination of voting rules is $\{\mathsf{Plurality}, \mathsf{STV}\}$. Here, we were unable to provide a black-box reduction showing that the combination of these rules is hard to manipulate. However, a careful inspection of Bartholdi and Orlin's proof [2] establishes that $\{\mathsf{Plurality}, \mathsf{STV}\}$-SM is indeed NP-hard: by tweaking the instance of STV constructed in that proof we can ensure that the manipulator's preferred candidate is the unique Plurality winner.

However, there are also examples where the combination of a hard-to-manipulate rule and an easy-to-manipulate one is easy to manipulate. Consider, for instance, the following rule: if some candidate receives strictly more than $\lfloor n/2 \rfloor$ first-place votes, he is the unique election winner; otherwise, all candidates are winners. We will refer to this rule as the Majority rule. Majority is not particularly decisive, but apart from that it is a reasonable voting rule. Clearly, it is easy to manipulate: the manipulator simply needs to check if ranking p first does the job. Moreover, the combination of Majority and STV is easy to manipulate, too.

THEOREM 4.4. $\{\mathsf{Majority}, \mathsf{STV}\}$-SM *is in* P.

PROOF. Consider an election $E = (C, V, \mathcal{R})$. If in this election p is ranked first by at most $\lfloor n/2 \rfloor - 1$ voters, the manipulator cannot make p the Majority winner, so this is a "no"-instance of our problem. On the other hand, if p is ranked first by at least $\lfloor n/2 \rfloor$ voters, the manipulator can rank p first, making him both the unique Majority winner and the unique STV winner. $\square$

The reason why the combination of Majority and STV is easy to manipulate is that Majority is always guaranteed to elect the STV winner: if some candidate has more than $\lfloor n/2 \rfloor$ votes, he will obviously win under STV, and in all other cases Majority elects all candidates. Using this observation, we can now generalize Theorem 4.4. We will say that a voting correspondence $\mathcal{F}_1$ is a *refinement* of a voting correspondence $\mathcal{F}_2$ if for any election E we have $\mathcal{F}_1(E) \subseteq \mathcal{F}_2(E)$, and there exists an election for which this containment is strict. Now, it is easy to see that STV is a refinement of Majority. Also, some of the voting rules defined in Section 2 are refinements of each other: namely, both Copeland and Maximin are refinements of Condorcet. Yet another example is provided by the so-called second-order Copeland rule, proved to be NP-hard to manipulate in [3]: this rule is obtained by combining the Copeland rule with a rather sophisticated tie-breaking rule, and is therefore a refinement of Copeland. Now, it is easy to see that the proof of Theorem 4.4 implies a more general fact.

COROLLARY 4.5. *If a voting correspondence $\mathcal{F}_1$ is a refinement of a voting correspondence $\mathcal{F}_2$ and $\mathcal{F}_2$-*SM *is in* P*, then so is* $\{\mathcal{F}_1, \mathcal{F}_2\}$-SM.

We remark that Corollary 4.5 crucially relies on the fact that we consider the unique-winner version of SM, and the requirement that a voting correspondence should produce a non-empty set of winners for every election. Also, the converse of Corollary 4.5 is not true, as illustrated by Copeland and second-order Copeland. Another important observation is that Corollary 4.5 applies equally well to the coalitional manipulation problem; we will make use of this fact in Section 5.

Now, suppose we have two hard-to manipulate rules. Clearly, it can be the case that their combination is also hard to manipulate: for example we can take two copies of STV (if we insist that these two rules should be distinct, we can modify one of the copies to produce a different winner on a single profile; this does not affect the complexity of our problem). To conclude this section, we provide an example of two voting rules $\mathcal{F}_1$ and $\mathcal{F}_2$ such that both $\mathcal{F}_1$-SM and $\mathcal{F}_2$-SM are NP-complete, but $\{\mathcal{F}_1, \mathcal{F}_2\}$-SM is in P; thus, counterintuitively, even a combination of hard-to-manipulate rules can be "easy" to manipulate (it will become clear in a minute why we used quotes in the previous sentence).

Our first voting rule is STV. Our second rule, which we will denote by STV$'$, is obtained from STV by the following modification: if c_i is the STV winner in E, then we output c_{i+1} as the unique winner (where $c_{m+1} := c_1$). Now, clearly, manipulating STV$'$ is just as hard as manipulating STV: we simply have to solve the STV manipulation problem for a different candidate. However, for any election E, STV and STV$'$ have different winners, so there is no way the manipulator can make p win under both of them. Thus, the manipulator's problem is "easy", in the sense that it simply cannot achieve its goal, so every instance of $\{\text{STV}, \text{STV}'\}$-SM is a "no"-instance. We summarize these observations as follows.

THEOREM 4.6. *STV$'$-SM is NP-complete. On the other hand, $\{\text{STV}, \text{STV}'\}$-SM is in P.*

We remark that Theorem 4.6 extends trivially to coalitional manipulation.

5. COALITIONAL MANIPULATION

The coalitional manipulation problem is known to be NP-hard for many prominent voting rules, such as Borda [6, 8] and some other scoring rules [20], Copeland$^\alpha$ for $\alpha \in (\mathbb{Q} \cap [0, 1]) \setminus \{0.5\}$ [11, 12] and Maximin [21]; it goes without saying that the hardness result for STV-SM [2] implies that STV-CM is NP-hard as well. Therefore, we cannot hope for a general easiness result along the lines of Theorem 4.1. Nevertheless, we can identify some interesting combinations of voting rules for which CM is in P.

We start by observing that Condorcet-CM is in P. Indeed, the manipulators can simply rank p first in all of their votes and check if that makes p the Condorcet winner; note that the answer to this question does not depend on how the manipulators rank the other candidates. Now, by extending Corollary 4.5 to the coalitional manipulation problem, and using the fact both Maximin and Copeland are refinements of the Condorcet rule, we obtain the following corollaries.

COROLLARY 5.1. *$\{\text{Condorcet}, \text{Maximin}\}$-CM is in P.*

COROLLARY 5.2. *$\{\text{Condorcet}, \text{Copeland}^\alpha\}$-CM is in P for any $\alpha \in \mathbb{Q} \cap [0, 1]$.*

Of course, the coalitional manipulation problem is also easy for the Majority rule, and it can be easily checked that each of the rules defined in Section 2 is a refinement of the Majority rule. Thus, we could obtain a similar easiness result for the combination of Majority and any other rule. We chose to state Corollaries 5.1 and 5.2 for the Condorcet rule, as the latter is more decisive and has been considered in prior work on computational social choice, albeit in the context of control [4].

We will now move on to another family of voting rules whose combinations can be shown to be easy to manipulate. A recent paper by Lin [18] shows that the coalitional manipulation problem is easy for k-Approval for any value of k. We will now prove a stronger statement: coalitional manipulation is easy even for combinations of k-Approval rules (for different values of k).

THEOREM 5.3. *For any finite set $K = \{k_1, \dots, k_\ell\} \subseteq \mathbb{N}$, the problem $\{k_1\text{-Approval}, \dots, k_\ell\text{-Approval}\}$-CM is in P.*

PROOF. Consider an election E with $C = \{c_1, \dots, c_m\}$, $|V| = h$, $|M| = s$, and $\mathcal{R} = (R_1, \dots, R_h)$. We can assume without loss of generality that $p = c_m$.

Since k-Approval is monotone for any value of k, it is optimal for the manipulators to rank p first in all s votes. For each $k \in K$, let $s_k(p)$ be p's k-Approval score in the resulting election. Now, the manipulators' goal is to rank every other candidate $c \in C \setminus \{p\}$ so that for each $k \in K$ the k-Approval score of c is strictly less than $s_k(p)$. We can assume without loss of generality that for each $k \in K$ and each $c \in C \setminus \{p\}$ the k-Approval score of c in $\mathcal{R}$ is strictly less than $s_k(p)$: otherwise, we clearly have a "no"-instance of our problem. Now, for each $r = 2, \dots, m$ and each c_j, $j = 1, \dots m-1$, let $x(r, j)$ be the maximum number of times that c_j can be ranked in position r or higher in the manipulators' votes so that its k-Approval score is less than $s_k(p)$ for every $k \in K$. These values are easy to compute from the candidates' k-Approval scores in $\mathcal{R}$, $k \in K$; our assumption on the initial scores ensures that they are non-negative.

We will now construct a flow network so that the maximum flow in this network corresponds to a successful set of manipulative votes, if one exists. Our network has a source S, a sink T, a node c_j for each $j = 1, \dots, m-1$, and a node p_r for $r = 2, \dots, m$; intuitively, node p_r corresponds to position r in the manipulators' votes. There is an edge of capacity s from S to each c_j, $j = 1, \dots, m-1$, and an edge of capacity s from each p_r, $r = 2, \dots, m$, to T. Essentially, the edge from S to c_j ensures that c_j is ranked by each manipulator, and the edge from p_r to T ensures that each of the manipulators fills position r in his vote. It remains to explain how to connect the candidates with the positions.

For each $c_j \in C \setminus \{p\}$ we build a caterpillar graph that connects c_j to $p_m, \dots, p_2$. More formally, for each candidate c_j, $j = 1, \dots, m-1$, we introduce nodes $z_{j,m}, \dots, z_{j,2}$ and edges $(c_j, z_{j,m})$, $(z_{j,r}, z_{j,r-1})$ for $r = m, \dots, 3$, and $(z_{j,r}, p_r)$ for $r = m, \dots, 2$. The capacity of $(c_j, z_{j,m})$ and $(z_{j,r}, p_r)$, $r = m, \dots, 2$, is $+\infty$, and the capacity of $(z_{j,r}, z_{j,r-1})$, $r = m, \dots, 3$, is given by $x(r-1, j)$. This completes the description of our network (see Figure 1).

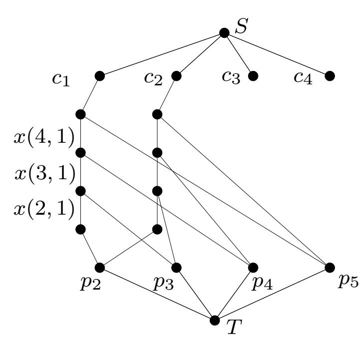

Figure 1: Network in the proof of Theorem 5.3, $m = 5$

We claim that this network admits a flow of size $s(m-1)$ if and only if there exists an assignment of candidates to the positions in the manipulators' votes such that the k-Approval score of each $c \in C \setminus \{p\}$ is less than $s_k(p)$ for every $k \in K$. Indeed, suppose that such a flow exists. Since all capacities are integer, we can assume that this flow is integer. It saturates all edges leaving S, so there are s units of flow leaving each c_j, $j = 2, \dots, m$. This flow has to reach $p_2, \dots, p_m$ traveling through the caterpillar graph associated with c_j. Thus, we can associate the flow on the edge $(z_{j,r}, p_r)$ with the number of times that c_j is ranked in position p_r. The capacity constraints on edges guarantee that these numbers correspond to a valid set of manipulators' votes. Moreover, for each $r = m, \dots, 2$, the total flow from c_j to $p_r, \dots, p_2$ is at most $x(r, j)$, which ensures that c_j is ranked in positions $p_r, \dots, p_2$ at most $x(r, j)$ times. Hence, for each $k \in K$ and each $j = 1, \dots, m-1$, the k-Approval score of c_j is less than that of p, and therefore p is the unique winner under each of the rules in our collection. Conversely, a vote that makes p the unique election winner with respect to each k-Approval, $k \in K$, can be converted into a valid flow; if x manipulators rank c_j in position r, we send x units of flow on $(z_{j,r}, p_r)$. $\square$

Theorem 5.3 has an interesting implication. Let $\widehat{\mathcal{F}}_\alpha$ be the family of all scoring rules. Observe that $\widehat{\mathcal{F}}_\alpha$ includes the Borda rule, for which coalitional manipulation is hard. Nevertheless, it turns out that $\widehat{\mathcal{F}}_\alpha$-CM is solvable in polynomial time.

THEOREM 5.4. *$\widehat{\mathcal{F}}_\alpha$-CM is in* P.

PROOF. We will use the following folklore observation [17]: a candidate c is the unique winner of an election $E = (C, V, \mathcal{R})$ with respect to each $|C|$-candidate scoring rule if and only if c is the unique k-Approval winner of E for $k = 1, \dots, |C| - 1$. Thus, to solve $\widehat{\mathcal{F}}_\alpha$-CM on an instance with m candidates, it suffices to apply the algorithm described in the proof of Theorem 5.3 with $K = \{1, \dots, m-1\}$. Clearly, the running time of this algorithm is polynomial in n and m. $\square$

We will now provide several examples of combinations of rules for which coalitional manipulation is hard. We will focus on classic voting rules, and investigate combinations of the most prominent easy-to-manipulate rule, namely, Plurality, with Borda and Copeland, which are both hard for coalitional manipulation.

THEOREM 5.5. *$\{$Plurality, Borda$\}$-CM is* NP*-complete.*

PROOF. Similarly to the proofs in Section 4, we will start with an instance of Borda-CM; this problem is known to be NP-hard even for two manipulators and three input votes [8]. Consider an instance of Borda-CM with $C = \{c_1, \dots, c_m\}$, $|V| = 3$, $\mathcal{R} = (R_1, R_2, R_3)$ and $|M| = 2$; assume without loss of generality that $m \geq 8$. Since both Borda and Plurality are neutral, we can assume without loss of generality that $p = c_m$.

For each $i = 1, \dots, m-1$, let X_i be an arbitrary vote in $\mathcal{L}(C)$ that ranks p first and c_i last, and let X'_i be the vote obtained by reversing X_i (i.e., if in X_i candidate c is ranked above candidate d, then in X'_i candidate d is ranked above c, for any $c, d \in C$). We modify the input election by adding votes X_i and X'_i for all $i = 1, \dots, m-1$; denote the resulting election by $\mathcal{R}'$. Note that this increases p's Plurality score by $m-1$, but the Plurality score of any other candidate only increases by 1. On the other hand, the Borda score of each candidate increases by exactly $(m-1)^2$.

Suppose we have started with a "yes"-instance of Borda-CM, and let L_1, L_2 be the manipulators' votes such that p is the unique Borda winner of $(\mathcal{R}, L_1, L_2)$. Clearly, p is also the unique Borda winner of $(\mathcal{R}', L_1, L_2)$. Moreover, the Plurality score of p in $(\mathcal{R}', L_1, L_2)$ is at least $m - 1 \geq 7$, while the Plurality score of any other candidate is at most $|V| + |M| + 1 = 6$, so p is also the unique Plurality winner in $(\mathcal{R}', L_1, L_2)$ (the careful reader will notice that we can relax the requirement that $m \geq 8$ by observing that Borda is monotone). Conversely, suppose that our instance of $\{$Plurality, Borda$\}$-CM is a "yes"-instance. Then there exist some votes $L'_1, L'_2 \in \mathcal{L}(C)$ that make p the unique Borda winner of $(\mathcal{R}', L'_1, L'_2)$. But then p is also the unique Borda winner of $(\mathcal{R}, L_1, L_2)$. $\square$

It is interesting to compare Theorem 5.4 and Theorem 5.5: the former implies that the combination of Borda with *all* k-Approval rules is easy to manipulate, whereas the latter shows that the combination of 1-Approval (i.e., Plurality) and Borda is hard to manipulate; we remark that the proof of Theorem 5.5 extends easily to the combination of Borda with k-Approval for any constant k.

A construction similar to the one used in the proof of Theorem 5.5 shows that $\{$Plurality, Copeland$^\alpha\}$-CM is NP-complete for $\alpha \in (\mathbb{Q} \cap [0,1]) \setminus \{0.5\}$ (this is the range of values of α for which Copeland$^\alpha$-CM in known to be NP-complete). The only difference is that for Copeland we cannot assume that the number of voters is a small constant (we will, however, assume that there are exactly two manipulators, as this is known to be sufficient for the NP-hardness of this problem [11, 12]). Therefore, instead of adding one pair (X_i, X'_i) for each $i = 1, \dots, m-1$, we add h such pairs, where h is the number of honest voters. This modification has no impact on Copeland scores: if c beats d in the original profile, this remains to be the case when the new votes are added; the converse is also true. However, the Plurality score of p increases by $h(m-1)$, whereas the Plurality score of any other candidate increases by h, and, as a result, does not exceed $2h + 2$ (even taking the manipulators' votes into account). Assuming without loss of generality that $m \geq 4$ and $h \geq 3$, we obtain that p is the unique Plurality winner of the modified election, irrespective of how the manipulator votes. The rest of the argument proceeds as in the proof of Theorem 5.5. We obtain the following corollary.

COROLLARY 5.6. *$\{$Plurality, Copeland$^\alpha\}$-CM is* NP*-complete for $\alpha \in (\mathbb{Q} \cap [0,1]) \setminus \{0.5\}$.*

Perhaps unsurprisingly, the combination of Borda and Copeland is hard to manipulate as well.

THEOREM 5.7. *$\{$Borda, Copeland$^\alpha\}$-CM is* NP*-complete for $\alpha \in (\mathbb{Q} \cap [0,1]) \setminus \{0.5\}$.*

PROOF. We employ a variant of the construction used in the proof of Corollary 5.6: we start with an instance of Copeland$^\alpha$-CM with $C = \{c_1, \dots, c_m\}$, $|V| = h$, $|M| = 2$, and $\mathcal{R} = (R_1, \dots, R_h)$ and modify it to obtain an instance of our problem in which p is the Borda winner no matter how the manipulator votes.

We assume without loss of generality that $h > 2$. Let $C' = C \cup \{d\}$, and modify $\mathcal{R}$ by ranking d in the last position in each preference order; denote the resulting profile by $\mathcal{R}'$. In $\mathcal{R}'$, d loses all pairwise elections, no matter how the manipulator votes; moreover, the final Copeland$^\alpha$ scores of all candidates do not depend on how the manipulators rank d.

Now, let X be some vote in $\mathcal{L}(C \cup \{d\})$ that ranks p first and d second, let X' be obtained by reversing X, and let X'' be obtained from X' by swapping p and d. Add $2mh$ pairs of the form (X, X'') to $\mathcal{R}'$; denote the resulting profile by $\mathcal{R}''$. Clearly, the addition of these new votes cannot possibly change the outcome of any pairwise election other than the one between p and d; moreover, p won his pairwise election against d even before these new votes were added, so this remains to be the case. We conclude that the Copeland$^\alpha$ score of any candidate $c \in C$ in $\mathcal{R}''$ exceeds his Copeland$^\alpha$ score in $\mathcal{R}$ by exactly 1.

	SM	CM
easy + easy = easy	all "nice" rules	k-Approval
easy + easy = hard	$\{\mathsf{STV}_1, \mathsf{STV}_2\}$	$\{\mathsf{STV}_1, \mathsf{STV}_2\}$
easy + hard = hard	$\{\mathsf{Borda}, \mathsf{STV}\}$, $\{\mathsf{Plurality}, \mathsf{STV}\}$	$\{\mathsf{Plurality}, \mathsf{Borda}\}$, $\{\mathsf{Plurality}, \mathsf{Copeland}\}$
easy + hard = easy	$\{\mathsf{Majority}, \mathsf{STV}\}$	$\{\mathsf{Condorcet}, \mathsf{Copeland}\}$, $\{\mathsf{Condorcet}, \mathsf{Maximin}\}$, scoring rules
hard + hard = easy	$\{\mathsf{STV}, \mathsf{STV}'\}$	$\{\mathsf{STV}, \mathsf{STV}'\}$
hard + hard = hard	$\{\mathsf{STV}, \mathsf{STV}\}$	$\{\mathsf{Borda}, \mathsf{Copeland}\}$

Table 1: Summary of results

On the other hand, the new votes increase the Borda score of every candidate other than p and d by $2m^2h$ (relative to $\mathcal{R}'$), whereas the Borda score of p goes up by $2m^2h + 2mh$ and Borda score of d goes up by $2m^2h - 2mh$. Since the Borda scores of all candidates in $\mathcal{R}'$ do not exceed mh, p is the unique Borda winner in $\mathcal{R}''$ irrespective of how the manipulator votes. The rest of the proof is similar to that of Theorem 5.5 and Corollary 5.6. $\square$

We remark that the proofs of Theorems 4.3, 5.7 and 5.5 and Corollary 5.6 are based on the same idea: we can modify an election so that the (relative) scores of all candidates with respect to one rule remain essentially unchanged while making a certain candidate a winner with respect to another voting rule. This suggests that these rules exhibit certain independence; this is somewhat reminiscent of Klamler's work on closeness of voting rules (see Klamler [15] and references therein). Formalizing this notion of independence is an interesting direction for future work.

6. CONCLUSIONS AND FUTURE WORK

We have investigated the problem of (coalitional) manipulation under uncertainty about the voting rules. Our results are summarized in Table 1. While we have not established the complexity of our problem for all possible combinations of voting rules, our results identify a number of approaches for dealing with problems of this type and the features of voting rules that make their combinations easy or hard to manipulate.

An obvious direction for future work is extending our approach to other forms of cheating in elections, such as control and bribery. Also, an interesting variant of our problem in the context of single-winner manipulation can be obtained by adopting the paradigm of safe strategic voting [19]. That is, instead of assuming that the manipulator wants to get a certain candidate elected, we take the more traditional approach, where the manipulator, too, has a preference order and would like to improve the election outcome with respect to this order; we can then ask whether the manipulator can vote so that the outcome improves for at least one voting rule in the given family and does not get worse with respect to the other rules.

Acknowledgments This research was supported by National Research Foundation (Singapore) under grant 2009-08. We would like to thank the anonymous AAMAS referees for their very useful feedback.

7. REFERENCES

[1] Y. Bachrach, N. Betzler, and P. Faliszewski. Probabilistic possible winner determination. In *AAAI'10*, pages 697–702, 2010.

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

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

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

[5] D. Baumeister, M. Roos, and J. Rothe. Computational complexity of two variants of the possible winner problem. In *AAMAS'11*, pages 853–860, 2011.

[6] N. Betzler, R. Niedermeier, and G. J. Woeginger. Unweighted coalitional manipulation under the Borda rule is NP-hard. In *IJCAI'11*, pages 55–60, 2011.

[7] V. Conitzer, T. Walsh, and L. Xia. Dominating manipulations in voting with partial information. In *AAAI'11*, pages 638–643, 2011.

[8] J. Davies, G. Katsirelos, N. Narodytska, and T. Walsh. Complexity of and algorithms for Borda manipulation. In *AAAI'11*, pages 657–662, 2011.

[9] E. Elkind, P. Faliszewski, and A. Slinko. Cloning in elections. In *AAAI'10*, pages 768–773, 2010.

[10] E. Elkind and H. Lipmaa. Hybrid voting protocols and hardness of manipulation. In *ISAAC'05*, pages 206–215, 2005.

[11] P. Faliszewski, E. Hemaspaandra, and H. Schnoor. Copeland voting: Ties matter. In *AAMAS'08*, pages 983–990, 2008.

[12] P. Faliszewski, E. Hemaspaandra, and H. Schnoor. Manipulation of Copeland elections. In *AAMAS'10*, pages 367–374, 2010.

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

[14] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. Hybrid elections broaden complexity-theoretic resistance to control. *Mathematical Logic Quarterly*, 55(4):397–424, 2009.

[15] C. Klamler. On the closeness aspect of three voting rules: Borda—Copeland—Maximin. *Group Decision and Negotiation*, 14:233–240, 2005.

[16] K. Konczak and J. Lang. Voting procedures with incomplete preferences. In *MPREF'05*, pages 124–129, 2005.

[17] J. Lang, 2010. Private communication.

[18] A. Lin. The complexity of manipulating k-approval elections. In *ICAART'11*, pages 212–218, 2011.

[19] A. Slinko and S. White. Non-dictatorial social choice rules are safely manipulable. In *COMSOC'08*, pages 403–413, 2008.

[20] L. Xia, V. Conitzer, and A. Procaccia. A scheduling approach to coalitional manipulation. In *ACM EC'10*, pages 275–284, 2010.

[21] L. Xia, M. Zuckerman, A. Procaccia, V. Conitzer, and J. Rosenschein. Complexity of unweighted manipulation under some common voting rules. In *IJCAI'09*, pages 348–353, 2009.