Hybrid Voting Protocols and Hardness of Manipulation

Edith Elkind[1] and Helger Lipmaa[2]

[1] Princeton University, Department of Computer Science,
35 Olden St, Princeton, NJ 08544, USA
[2] Cybernetica AS and University of Tartu, Estonia

Abstract. This paper addresses the problem of constructing voting protocols that are hard to manipulate. We describe a general technique for obtaining a new protocol by combining two or more base protocols, and study the resulting class of (vote-once) hybrid voting protocols, which also includes most previously known manipulation-resistant protocols. We show that for many choices of underlying base protocols, including some that are easily manipulable, their hybrids are NP-hard to manipulate, and demonstrate that this method can be used to produce manipulation-resistant protocols with unique combinations of useful features.

1 Introduction

In multiagent systems, the participants frequently have to agree on a joint plan of action, even though their individual opinions about the available alternatives may vary. Voting is a general method of reconciling these differences, and having a better understanding of what constitutes a good voting mechanism is an important step in designing better decision-making procedures. In its most general form, a voting mechanism is a mapping from a set of votes (i.e., voters' valuations for all alternatives) to an ordering of the alternatives that best represents the collective preferences. In many cases, however, the attention can be restricted to mechanisms that interpret their inputs (votes) as total orderings of the alternatives/candidates and output a single winner. A classical example here is Plurality voting, where only the top vote of each voter is taken into account, and the candidate with the largest number of top votes wins.

A fundamental problem encountered by all voting mechanisms is *manipulation*, i.e., the situation when a strategizing voter misrepresents his preferences in order to obtain a more desirable outcome. One can expect that rational agents will engage in manipulation whenever it is profitable for them to do so; as a result, the output of the voting mechanism may grossly misrepresent the actual preferences of the agents and be detrimental to the system as a whole.

It is well-known [8, 11] that any nondictatorial voting mechanism for three or more candidates is susceptible to manipulation. However, while there is no information-theoretic solution to this problem, one can try to discourage potential manipulators by making manipulation infeasible. This approach is particularly attractive in multiagent setting, when decisions have to be made in real time, and whether an agent can find a beneficial manipulation quickly is more important than whether such a manipulation exists in principle. It turns out that some of the voting protocols that are used in practice enjoy this property: it has been shown [1, 2] that second-order Copeland and Single Transferable Vote (STV) are NP-hard to manipulate. Furthermore, in a recent paper [4], Conitzer and Sandholm showed that several protocols, including Borda, STV, Maximin and Plurality, can be modified so that manipulating them becomes computationally hard. Their method involves prepending the original protocol by a pre-round in which candidates are divided into pairs and the voters' preferences are used to determine the winner of each pair; the winners of the pre-round participate in elections conducted according to the original protocol. Different methods for pairing up the candidates and eliciting the votes give rise to different levels of complexity, such as NP-hardness, #P-hardness, or PSPACE-hardness.

The advantage of this method of constructing manipulation-resistant protocols is in preserving some of the properties of the original protocol: for example, if the base protocol is Condorcet-consistent (see Section 6 for definition), then the modified protocol is Condorcet-consistent as well. However, for some other desirable features this is not true, and, generally, eliminating half of the candidates using a set of criteria that may be very different in spirit from those used by the original protocol, is likely to alter the outcome considerably, so that the desiderata that motivated the original protocol may no longer be attainable.

We build upon the ideas of [4] to construct a larger family of protocols that are hard to manipulate. We observe that their pre-round phase can be viewed as the first stage of the voting protocol known as Binary Cup (BC) (defined in Section 2). While this protocol itself is not hard to manipulate (at least, when the schedule is known in advance), the results of [4] can be interpreted as showing that combining BC with other protocols results in manipulation-resistant schemes. We generalize this idea by showing that this kind of hardness amplification is not unique to BC.

We define the class of *(vote-once) hybrid voting protocols* $\mathsf{Hyb}(\mathsf{X}_k, \mathsf{Y})$. In $\mathsf{Hyb}(\mathsf{X}_k, \mathsf{Y})$, after the voters have expressed their preferences, k steps of protocol X are performed to eliminate some of the candidates, and then protocol Y is run on the rest of the candidates, reusing the votes as restricted to the remaining candidates. In practice, such a reuse of votes is important, since it allows voters to only express their preferences once; this feature is desirable both for actual elections, where it is difficult to get citizens to the voting booths more than once, and for artificial agents, where round complexity of a protocol may be an issue. Clearly, the protocols of [4] belong to this family, as does STV; therefore, our framework encompasses most of the known hard-to-manipulate voting mechanisms.

We show that many other hybrid protocols are NP-hard to manipulate as well. Specifically, we consider several well-known protocols, such as Plurality, Borda, STV, and Maximin, and prove that many hybrids of these protocols are manipulation-resistant. We do this by formulating some fairly general conditions on X and Y under which the protocols of the form $\mathsf{Hyb}(\mathsf{X}_k, \mathsf{Plurality})$, $\mathsf{Hyb}(\mathsf{X}_k, \mathsf{STV})$, or $\mathsf{Hyb}(\mathsf{STV}_k, \mathsf{Y})$ are NP-hard to manipulate. Additionally, we show that a hybrid of a protocol with itself may be different from the original protocol — and much harder to manipulate. We prove that this is, indeed, the case for Borda protocol: $\mathsf{Hyb}(\mathsf{Borda}_k, \mathsf{Borda})$ is NP-hard to manipulate, while Borda itself is easily manipulable.

We define a generic closure operation on protocols that makes them closed under hybridization. Interestingly, applying this operation to the easy-to-manipulate Plurality results in the hard-to-manipulate STV. We conjecture that for many other basic protocols, their closed versions are NP-hard to manipulate as well. Whenever this is the case, the closed protocols provide the most faithful manipulation-resistant approximation to the underlying protocols, which makes them compelling alternatives to the original protocols.

On the flip side, we demonstrate that hybridization does not always result in hard-to-manipulate protocols: in particular, the hybrid protocols that use Plurality as their first component, are almost as easy to manipulate as their second component. Finally, we demonstrate that our techniques extend to voting protocols that allow voters to rate the candidates rather than just order them.

The value of our results is not so much in constructing specific new manipulation-resistant protocols, but rather in providing a general method for doing that, which can be used with many basic schemes. Since a hybrid inherits some of the properties of its ingredients, we get hard-to-manipulate protocols with properties not shared by the schemes from [1, 2, 4]. For example, since BC is not Pareto-optimal, all protocols obtained by the method of [4] are not Pareto-optimal either, while our approach allows to construct hybrids that have this valuable feature (for definitions, see Section 6). It has already been argued in [4] that it is desirable to have manipulation-resistant protocols that can be used in different real-life situations; our method fits the bill.

The use of voting and voting-related techniques is not restricted to popular elections: the ideas from this domain have been applied in rank aggregation [5, 9], recommender systems [10], multiagent decision making in AI [7], etc. In many of these settings, the number of alternatives is large enough to make our results applicable, and, furthermore, the agents are both sufficiently sophisticated to attempt manipulation and may derive significant utility from doing so. Therefore, we feel that it is important to have a better understanding of what makes voting protocols hard to manipulate, as this will allow us to design more robust decision-making systems that use voting-like methods.

The rest of the paper is organized as follows. In Section 2 we introduce our notation, give a precise definition of what it means to manipulate an election, and describe some well-known voting schemes discussed in the paper. In Section 3, we define hybrid protocols and some related notions. In Section 4, we show that certain hybrid protocols are NP-hard to manipulate. In Section 5, we discuss hybrids obtained by combining a protocol with itself. In Section 6, we define some desirable properties of voting protocols, show that many of them are preserved under hybridization, and demonstrate that our protocols can provide useful combinations of these properties. In Section 7, we provide examples of hybrids that are easy to manipulate and discuss limitations and extensions of our approach. Finally, in Section 8, we present our conclusions and future research directions.

2 Preliminaries and Notation

We assume that there are n voters and m candidates and denote the set of all voters by $V = \{v_1, \dots, v_n\}$ and the set of all candidates by $C = \{c_1, \dots, c_m\}$. Most of our complexity results are in terms of m and n, i.e., unless specified otherwise, 'polynomial' always means 'polynomial in m and n'.

The set of all permutations of C is denoted by $\Pi(C)$; the preference of the ith voter is expressed by a list $\pi_i \in \Pi(C)$: the first element is the voter's most preferred candidate, etc. In particular, this means that within one voter's preference list, ties are not allowed. For any subset $C' \subseteq C$, let $\pi|_{C'}$ be the permutation π as restricted to C' (i.e., elements not from C' are omitted). Note that $\pi|_{C'}$ corresponds to a valid preference in an election that has the candidate set C'.

When describing the preferences of a single voter v, we write $c_i \succ_v c_j$ to denote that v prefers c_i to c_j. Similarly, we write $C_i \succ_v C_j$ to denote that v prefers all candidates in the set C_i to all candidates in the set C_j, without specifying the ordering of candidates within C_i and C_j. When the identity of the voter is clear from the context, we omit the subscript and write $\succ$ instead of $\succ_v$.

A *voting protocol* is a mapping $P : \langle \Pi(C), \dots, \Pi(C) \rangle \mapsto C$ that selects a winner $c \in C$ based on all voters' preference lists. In this paper, we consider the following common voting protocols (in all definitions that mention points, the candidate with the most points wins):

Plurality: A candidate receives 1 point for every voter that ranks it first.

Borda: For each voter, a candidate receives $m - 1$ point if it is the voter's top choice, $m - 2$ if it is the second choice, ..., 0 if it is the last.

Single Transferable Vote (STV): The winner determination process proceeds in rounds. In each round, a candidate's score is the number of voters that rank it highest among the remaining candidates, and the candidate with the lowest score drops out. The last remaining candidate wins. (A vote transfers from its top remaining candidate to the next highest remaining candidate when the former drops out.)

Maximin: A candidate's score in a pairwise election is the number of voters that prefer it over the opponent. A candidate's number of points is the lowest score it gets in any pairwise election.

Binary Cup (BC): The winner determination process consists of $\lceil \log m \rceil$ rounds. In each round, the candidates are paired; if there is an odd number of candidates, one of them gets a bye. The candidate that wins the pairwise election between the two (or got a bye) advances into the next round. The schedule of the cup (i.e., which candidates face each other in each round) may be known in advance (i.e., before the votes are elicited) or it may depend on the votes.

Voting manipulation We say that a voter v_j can *manipulate* a protocol P if there is a permutation $\pi'_j \in \Pi(C)$ such that for some values of $\pi_i \in \Pi(C)$, $i = 1, \dots, n$, we have

1. $P(\pi_1, \dots, \pi_n) = c$;
2. $P(\pi_1, \dots, \pi_{j-1}, \pi'_j, \pi_{j+1}, \dots, \pi_n) = c' \neq c$;

3. v_j ranks c' above c.

We say that v_j manipulates P *constructively* if v_j ranks c' first and *destructively* otherwise. All results in this paper are on constructive manipulation; in what follows, we omit the word 'constructive'. A voter v_j manipulates P *efficiently* if there is a polynomial time algorithm that given preference lists $\pi_1, \dots, \pi_n$ for which such π'_j exists, constructs π'_j.

3 Hybrid Protocols

In this section, we formally define *(vote-once) hybrid protocols*. Intuitively, a hybrid of two protocols X and Y executes several steps of X to eliminate some of the candidates, and then runs Y on the remaining set of candidates. To make this intuition precise, however, we have to define how to interpret the first protocol X as a sequence of steps. While there is no obvious way to do this for an arbitrary protocol, most well-known protocols, including the ones described in Section 2, admit such an interpretation. In particular, we suggest the following definitions:

- For STV, a *step* is a single stage of the protocol. That is, a step of STV consists of eliminating a candidate with the least number of first-place votes and transferring each vote for this candidate to the highest remaining candidate on that ballot.
- For $\mathsf{Binary\ Cup}$ (BC), a *step* is a single stage of the protocol as well, i.e., it consists of pairing up the candidates and eliminating the ones who lose in the pairwise comparison.
- For point-based protocols, such as $\mathsf{Plurality}$, Borda, or $\mathsf{Maximin}$, we first compute the scores of all candidates, order them by their scores from the lowest to the highest, and define a *step* to consist of eliminating the first (i.e., the lowest ranked) remaining candidate in this sequence. Note that the scores are not recomputed between the steps. (A similar approach can be applied to any voting protocol that can be extended to a preference aggregation rule, i.e., a function that maps votes to total orderings of the candidates. In this case, the order in which the candidates are eliminated is obtained by inverting the output of the preference aggregation rule.)

Definition 1. *A* hybrid protocol $\mathsf{Hyb}(\mathsf{X}_k, \mathsf{Y})$ *consists of two* phases*. Suppose that the voters' preference lists are described by the n-tuple $(\pi_1, \dots, \pi_n)$. In the first phase, the protocol executes k steps of $\mathsf{X}(\pi_1, \dots, \pi_n)$; suppose that S is the set of candidates not eliminated in the first phase. In the second phase, the protocol applies Y to $(\pi_1|_S, \dots, \pi_n|_S)$, i.e., the preference lists restricted to the remaining set S of candidates.*

It is straightforward to extend this definition to hybrids $\mathsf{Hyb}(\mathsf{X}^{(1)}_{k_1}, \mathsf{X}^{(2)}_{k_1}, \dots, \mathsf{X}^{(t)}_{k_t}, \mathsf{Y})$ of three or more protocols.

4 Hardness Results

4.1 Hardness of STV-based Hybrids

In this subsection, we show that hybrids $\mathsf{Hyb}(\mathsf{STV}_k, \mathsf{Y})$ and $\mathsf{Hyb}(\mathsf{X}_k, \mathsf{STV})$ are NP-hard to manipulate for many "reasonable" voting protocols X and Y, including the cases $\mathsf{X}, \mathsf{Y} \in \{\mathsf{Plurality}, \mathsf{Borda}, \mathsf{Maximin}, \mathsf{BC}\}$.

Theorem 1. *A hybrid of the form $\mathsf{Hyb}(\mathsf{STV}_k, \mathsf{Y})$ is NP-hard to manipulate for infinitely many values of k as long as Y satisfies the following condition: Whenever there is a candidate c who receives K first-place votes and $n - K$ second-place votes, while all other candidates receive at most $K - 1$ first-place vote, Y declares c the winner.*

Proof. Bartholdi and Orlin show in [2] that STV is NP-hard to manipulate for infinitely many pairs (V', C'), where $V', |V'| = n'$, is the set of voters and $C', |C'| = m'$, is the set of candidates. Note that $n' \geq 2$. Let p be the manipulator's preferred candidate in this proof. For an arbitrary $t \geq 1$, we construct a new set of candidates $C = C' \cup \{c_1, \ldots, c_t\}$ and a new set of voters $V = V' \cup V_1 \cup \cdots \cup V_t$ with $|V_i| = n' - 1$. Define the new set of preferences by requiring that all $n' - 1$ honest voters in V' rank C' above $\{c_1, \ldots, c_t\}$, while the $n' - 1$ voters in V_i, $i = 1, \ldots, t$, rank c_i first and p second. Set $k = m' - 1$. We show that for this set of preferences, manipulating STV is exactly as hard as manipulating $\mathsf{Hyb}(\mathsf{STV}_k, \mathsf{Y})$. Then, due to the results of [2], it is NP-hard to manipulate $\mathsf{Hyb}(\mathsf{STV}_k, \mathsf{Y})$ for infinitely many pairs (V', C'), and for infinitely many values of k.

First, suppose that a voter $v \in V'$ can successfully manipulate STV for the pair (V', C') and for some preference lists of the honest voters. Then, if v extends his vote by ranking C' above $\{c_1, \ldots, c_t\}$, this new vote constitutes a successful manipulation of $\mathsf{Hyb}(\mathsf{STV}_k, \mathsf{Y})$ with the preference lists defined in the previous paragraph. Indeed, in the jth step of the STV phase, there is always a candidate in C' who has at most $n'/(m' - j + 1) \leq n'/2$ first-place votes, and each c_i has $n' - 1$ first-place vote, so all c_i survive the first phase. Hence, the set of candidates eliminated by the STV phase of the new protocol coincides with the set of candidates eliminated by STV in the construction of [2]. Thus, after $m' - 1$ rounds, p is ranked first by all n' voters in V'. By our condition on Y, this means that $\mathsf{Hyb}(\mathsf{STV}_k, \mathsf{Y})$ declares p the winner.

For the opposite direction, suppose that a voter v can manipulate $\mathsf{Hyb}(\mathsf{STV}_k, \mathsf{Y})$. This means that p survives the STV phase, and we have seen that the candidates $\{c_1, \ldots, c_t\}$ do not affect the execution of the STV phase. All that remains to show is that v's vote in these elections can be interpreted as a vote in STV. This is obviously the case if the manipulator ranks C' above $\{c_1, \ldots, c_t\}$. On the other hand, if he ranks some c_i above some candidates in C', as soon as all candidates that he ranks above c_i are eliminated, his vote is effectively dropped from the counting process. Now, the reduction of [2] has the following property: for any partial vote of the manipulator, i.e., a vote that only ranks a subset of C' and is discarded as soon as the last candidate from C' is eliminated, there exists an equivalent regular vote, i.e., a full ordering of C', that results in the same order of elimination. Therefore, we can convert the manipulator's vote into a successful manipulation of the original protocol. □

Corollary 1. *The hybrids* $\mathsf{Hyb}(\mathsf{STV}_k, \mathsf{Y})$, *where* $\mathsf{Y} \in \{\mathsf{Plurality}, \mathsf{Borda}, \mathsf{Maximin}, \mathsf{BC}, \mathsf{STV}\}$, *are NP-hard to manipulate for infinitely many values of* k.

The proof of this corollary is straightforward since all these voting protocols satisfy the required property.

Theorem 2. *A hybrid of the form* $\mathsf{Hyb}(\mathsf{X}_k, \mathsf{STV})$ *is NP-hard to manipulate for infinitely many values of* k *if* X *satisfies the following condition for some unbounded nondecreasing function* $f(\cdot)$ *and infinitely many* K*: Suppose that all but one voter rank some* K *candidates* $c_1, \ldots, c_K$ *after all other candidates, and all other candidates receive at least 2 first-place votes. Then after* $f(K)$ *steps of* X*, the set of eliminated candidates is a subset of* $\{c_1, \ldots, c_K\}$.

Proof (Sketch). Set $k = f(K)$. Denote the set of candidates in the construction of [2] by C'; let $C'' = \{c_1, \ldots, c_K\}$ and $C = C' \cup C''$. Modify the votes of all honest voters in that construction so that they rank C' above C''. The reduction of [2] has the property that each candidate in C' gets more than 2 first-place votes. Hence, the set of candidates eliminated in k rounds of X is a subset of C''; furthermore, the remaining candidates from C'' will be the first candidates eliminated by STV. Hence, no matter how the manipulator ranks the candidates in C'', it has no effect on the execution of the protocol. Therefore, his vote can be interpreted as a vote in the original STV and vice versa. □

Corollary 2. *The hybrids of the form* $\mathsf{Hyb}(\mathsf{X}_k, \mathsf{STV})$, *where* $\mathsf{X} \in \{\mathsf{Plurality}, \mathsf{Borda}, \mathsf{Maximin}, \mathsf{BC}\}$, *are NP-hard to manipulate for infinitely many values of* k.

Proof. It is easy to see that Plurality, Maximin and BC satisfy the condition of the theorem. For Borda, it is satisfied whenever the number of voters exceeds the number of candidates; in the construction of [2], the number of voters is larger than $3|C'|$, so we can set $K = |C'|$. □

Our proofs that hybrids using STV as their first or second component are NP-hard to manipulate rely on some specific properties of the reduction constructed in [2]. In the full version of the paper, we provide black-box constructions, i.e., ones that work with any NP-hardness proof.

4.2 Hybrids of the Form Hyb(X_k, Plurality)

In this subsection, we prove that Hyb(X_k, Plurality) is hard to manipulate whenever X satisfies Property 1. While this property might seem artificial, we show that it is possessed by at least two well-known protocols, namely, Borda and Maximin.

Property 1. For any set $G = \{g_1, \dots, g_N\}$, any collection $S = \{s_1, \dots, s_M\}$ of subsets of G, and any $K \leq M$, there are some k', $k' \leq M$, and T, $T > 3N$, such that it is possible to construct in polynomial time a set of $T + N(T-2) + 3N$ votes over the set of candidates $C' \cup C'' \cup \{p\}$, where $C' = \{c'_1, \dots, c'_N\}$, $C'' = \{c''_1, \dots, c''_M\}$, so that

- there are T voters who rank p first;
- for each $i = 1, \dots, N$, there are $T - 2$ voters who rank c_i first;
- for each $i = 1, \dots, N$, there are 3 voters who rank all c''_j such that $g_i \in s_j$ above c_i, and rank c_i above all other candidates;
- for any additional vote π, when it is tallied with all other votes, the set of candidates eliminated in the first k' rounds is a subset of C'' of size $M - K$;
- for any subset $S' \subseteq S$, $|S'| = M - K$, one can design in polynomial time a vote $\pi_{S'}$ that, when tallied with other votes, guarantees that the set of candidates eliminated in the first k' rounds is exactly $\{c''_i \mid s_i \in S'\}$.

Theorem 3. *A hybrid of the form* Hyb(X_k, Plurality) *is NP-hard to manipulate for infinitely many values of k whenever X satisfies Property 1.*

Proof. We give a reduction that is based on the NP-hard problem SET COVER. Recall that SET COVER can be stated as follows: Given a ground set $G = \{g_1, \dots, g_N\}$, a collection $S = \{s_1, \dots, s_M\}$ of subsets of G, and an integer K, does there exist a K-cover of G, i.e., a subset S' of S, $S' = \{s_1, \dots, s_K\}$, such that for every $g_i \in G$ there is an $s_j \in S'$ such that $g_i \in s_j$?

Construct the set of votes based on G, S, and K so that it satisfies Property 1. Let $k = k'$, and let p be the manipulator's preferred candidate. We show that the manipulator can get p elected under Hyb(X_k, Plurality) if and only if he can find a set cover for G. Indeed, after k rounds of X, all candidates in $C' \cup \{p\}$ survive, as well as exactly K candidates from C''. We show that p wins if and only if these K candidates correspond to a set cover of G. Observe that any surviving candidate from C'' has at most $3N < T$ first-place votes, so he cannot win in the last stage. Now, consider a candidate $c'_i \in C'$. Suppose that the corresponding element is not covered, i.e., all c''_j such that $g_i \in s_j$ are eliminated. Then after the end of the first phase, c'_i has $T + 1$ first-place vote, while p has T first-place votes, so in this case p cannot win.

On the other hand, suppose that for any $g_i \in G$ there is an $s_j \in S$ such that $g_i \in s_j$ and c''_j is not eliminated in the first phase. Then at the beginning of the second phase each $c'_i \in C'$ has $T - 2$ first-place votes, while p has T first-place votes, so in this case p wins.

Hence, manipulating this protocol is equivalent to finding a set cover of size K. □

Corollary 3. *The protocols* Hyb(Borda_k, Plurality) *and* Hyb($\mathsf{Maximin}_k$, Plurality) *are NP-hard to manipulate for infinitely many values of k.*

Proof. Let the voters who rank p first, rank the candidates in C' above those in C'', and the voters who rank c'_i first, rank the candidates in $p \cup C'$ above those in C''. For large enough T, this guarantees that both Borda and $\mathsf{Maximin}$ scores of the candidates in $C' \cup \{p\}$ are much higher than those of the candidates in C'', so none of the candidates in $C' \cup \{p\}$ can be eliminated in the first phase. On the other hand, we still have enough flexibility to ensure that all candidates in C'' have the same Borda (or $\mathsf{Maximin}$) score with respect to the honest voters' preferences. Then, for both protocols, the manipulator can get any $M - K$ candidates from C'' eliminated by putting them on the bottom of his vote and ranking the remaining K candidates above the candidates in $C' \cup \{p\}$. Thus, both Borda and $\mathsf{Maximin}$ satisfy all conditions in the statement of Theorem 3. □

Together with our results on STV and the results of [4], the constructions of this section provide a wide choice of manipulation-resistant protocols. In the next section, we add to our repertoire two more protocols that are hard to manipulate, namely, $\mathsf{Hyb}(\mathsf{Borda}_k, \mathsf{Borda})$ and $\mathsf{Hyb}(\mathsf{Maximin}_k, \mathsf{Borda})$.

5 Hybrid of a Protocol with Itself

We say that a protocol is *hybrid-proof* if a hybrid of several copies of this protocol is equivalent to the original protocol. While some protocols, such as STV or $\mathsf{Binary\ Cup}$, have this property, for many other protocols, especially score-based ones, this is not the case. To see this, note that in a hybrid protocol, the scores of all surviving candidates are recomputed in the beginning of the second phase, while in the original protocol they are computed only once. As a result, in a hybrid of, say, two copies of the $\mathsf{Plurality}$ protocol, one candidate may gain a lot of first-place votes from voters who rank him right after the candidates that were dropped in the first phase, while some other candidate may get no extra votes at all; a similar phenomenon happens in Borda and $\mathsf{Maximin}$.

Nevertheless, any protocol can be modified to be hybrid-proof. For an arbitrary protocol X, define a *closed protocol* $\overline{\mathsf{X}}$ by $\overline{\mathsf{X}} = \mathsf{Hyb}(\mathsf{X}_1, \dots, \mathsf{X}_1)$, where the number of copies of X_1 is such that $\overline{\mathsf{X}}$ selects a single winner.

Proposition 1. *For any protocol* X*, the closed protocol* $\overline{\mathsf{X}}$ *is hybrid-proof.*

We omit the proof.

Interestingly, $\mathsf{Hyb}(\mathsf{Plurality}_1, \dots, \mathsf{Plurality}_m) = \mathsf{STV}$: the vote transfer mechanism can be viewed as recomputing each candidate's $\mathsf{Plurality}$ score. Observe that while $\mathsf{Plurality}$ has particularly bad manipulation resistance properties (see, e.g., Section 7), STV is NP-hard to manipulate. This leads us to conjecture that for many other base protocols, the new protocols obtained in this manner are NP-hard to manipulate. Whenever this is the case, the closed protocols provide the most faithful manipulation-resistant approximation to the underlying protocols, which makes them compelling alternatives to the original protocols. This conjecture is supported by the fact that for some easy-to-manipulate protocols, a hybrid of just two copies of the protocol is NP-hard to manipulate; increasing the number of copies should make the manipulation harder, not easier. As an illustration, we prove that a hybrid of two instances of Borda is NP-hard to manipulate.

Theorem 4. *The hybrid* $\mathsf{Hyb}(\mathsf{Borda}_k, \mathsf{Borda})$ *is NP-hard to manipulate for infinitely many values of* k*.*

Proof. We give a reduction from EXACT COVER BY 3-SETS, which is stated as follows: Given a ground set $G = \{g_1, \dots, g_N\}$, $N = 3L$, and a collection $S = \{s_1, \dots, s_M\}$ of 3-element subsets of G, does there exist an exact set cover of G, i.e., a subset S' of S, $S' = \{s_1, \dots, s_{N/3}\}$ such that for every $g_i \in G$ there is a unique $s_j \in S'$ such that $g_i \in s_j$?

We construct two sets of voters V', $|V'| = 2N + 2$, and V'', $|V''| = (M + 1)(N + 1)$ and define $V = V' \cup V''$. Let $C^g = \{c^g_1, \dots, c^g_N\}$ and $C^s = \{c^s_1, \dots, c^s_M\}$, and let the set of candidates be $C = C^g \cup C^s \cup \{c_0\} \cup p$, where p is the manipulator's preferred candidate.

For each $i = 1, \dots, N-1$, the voters $v'_{2i-1}, v'_{2i} \in V'$ rank the candidates as

$$c^g_{i+1} \succ c^g_{i+2} \succ \dots \succ c^g_N \succ p \succ c^g_1 \succ \dots c^g_{i-1} \succ$$

$$\succ C^s_i \succ c^g_i \succ C^s \setminus C^s_i \succ c_0 \ ,$$

where $C^s_i = \{c^s_j \mid g_i \in s_j\}$. The voters $v'_{2N-1}, v'_{2N} \in V'$ rank the candidates as

$$p \succ c^g_1 \succ c^g_2 \succ \dots \succ c^g_{N-1} \succ C^s_N \succ c^g_N \succ C^s \setminus C^s_N \succ c_0 \ ,$$

where $C^s_N = \{c^s_j \mid g_N \in s_j\}$.

The remaining two voters in V' rank the candidates as

$$c^g_1 \succ c^g_2 \succ \dots \succ c^g_N \succ c_0 \succ p \succ C^s$$

and

$$c^g_1 \succ c^g_2 \succ \dots \succ c^g_N \succ p \succ c_0 \succ C^s.$$

Also, for each $i = 1, \dots, N-1$, there are $M+1$ voters in V'' who rank the candidates as

$$c^g_{i+1} \succ c^g_{i+2} \succ \dots \succ c^g_N \succ p \succ c^g_1 \succ \dots c^g_i \succ c_0 \succ C^s \ ,$$

$M+1$ voters in V'' who rank the candidates as

$$p \succ c^g_1 \succ c^g_2 \succ \dots \succ c^g_N \succ c_0 \succ C^s \ ,$$

and $M+1$ voters in V'' rank the candidates as

$$c^g_1 \succ c^g_2 \succ \dots \succ c^g_N \succ p \succ c_0 \succ C^s.$$

Set $k = M - N/3$. Observe that no matter how the manipulator votes, only the candidates from C^s will be eliminated in the first phase. All candidates in $C^g \cup \{p\}$ have the same Borda score with respect to V''. Furthermore, since we have not yet specified the preferences of voters in V'' over the candidates in C^s, we can set them so that they will all have the same Borda score, in which case the manipulator can get any k of them eliminated in the first phase.

Suppose that the manipulator votes so that the set of candidates from C^s who survive the first phase corresponds to an exact set cover of G. Then for each candidate c^g_i and any $j = 1, \dots, N$, there are two voters in V' who rank him in the jth position and two voters in V' who rank him in the $(N+2)$nd position (these two voters prefer c^s_j to c^g_i, where s_j is the set in the set cover that contains g_i). Hence, the Borda score of each candidate in C^g with respect to V' is $\sum_{t=m-k-N}^{m-k-1} 2t + 2(m-k-N-2)$.

On the other hand, the Borda score of p with respect to V' is $\sum_{t=m-k-N}^{m-k-1} 2t + (m-k-N-1) + (m-k-N-2)$, and the score of c_0 is lower that the score of any candidate in $C^g \cup \{p\}$, so in this case p wins.

Conversely, suppose that the set of candidates from C^s who survive the first phase does not correspond to a set cover of G. Consider an element $g_i \in G$ that is not covered. All voters in V' prefer c^g_i to all surviving candidates in $C^s \cup \{c_0\}$, which means that his Borda score is higher than that of p. □

Using the same construction, one can show that $\mathsf{Hyb}(\mathsf{Maximin}_k, \mathsf{Borda})$ is NP-hard to manipulate for infinitely many values of k; we omit the details.

6 Properties of Voting Protocols

Voting protocols are evaluated based on various criteria, such as:

(1) *Pareto-optimality*: a candidate who is ranked lower than some other candidate by every voter never wins;
(2) *Condorcet-consistency*: if there is a candidate who is preferred to every other candidate by a majority of voters, this candidate should be the winner of the election;
(3) *Monotonicity*: with the relative order of the other candidates unchanged, ranking a candidate higher should never cause the candidate to lose, nor should ranking a candidate lower ever cause the candidate to win.

In the context of this paper, a natural addition to this list is *hardness of manipulation*.

Most voting schemes based on pairwise comparisons, in particular, BC and $\mathsf{Maximin}$, are Condorcet-consistent, while for STV, or positional methods, such as $\mathsf{Plurality}$ or Borda, this is not the case. One can prove that $\mathsf{Plurality}$, Borda, $\mathsf{Maximin}$, and BC are monotone, while STV is not. All basic voting protocols considered in this paper except BC are Pareto-optimal.

To analyze whether properties (1)–(3) are preserved under hybridization, we have to extend these definitions to multi-step protocols. We say that a multi-step protocol is *strongly Pareto-optimal* if whenever every voter ranks c_1 below c_2, c_1 is eliminated before c_2, and *strongly monotone* if ranking a candidate higher does not affect the relative order of elimination of other candidates and cannot result in him being eliminated at an earlier step; the definition of Condorcet consistency remains unchanged. It is easy to see that multi-step versions of Pareto-optimal protocols that we consider are strongly Pareto-optimal, at least for some draw resolution rules. However, not all monotone protocols are strongly monotone: for example, in Borda, moving a candidate several positions up changes other candidates' scores in a non-uniform way.

Proposition 2. *For any voting protocols X and Y and any k, if both X and Y are Condorcet-consistent, so is $\mathsf{Hyb}(\mathsf{X}_k, \mathsf{Y})$; if X is strongly Pareto-optimal (strongly monotone) and Y is Pareto-optimal (monotone), then $\mathsf{Hyb}(\mathsf{X}_k, \mathsf{Y})$ is Pareto-optimal (monotone).*

We omit the proofs.

The construction proving that BC is not Pareto-optimal can be easily modified to show that any protocol of the form $\mathsf{Hyb}(\mathsf{BC}_k, \mathsf{Y})$ is not Pareto-optimal for some k, where $\mathsf{Y} \in \{\mathsf{Plurality}, \mathsf{Borda}, \mathsf{Maximin}, \mathsf{STV}\}$. Hence, prior to this work, the only Pareto-optimal mechanisms that were known to be NP-hard to manipulate were STV and the (rather contrived) variants of the Copeland protocol that were described in [1]. Our results imply that $\mathsf{Hyb}(\mathsf{Borda}_k, \mathsf{Plurality})$, $\mathsf{Hyb}(\mathsf{Maximin}_k, \mathsf{Plurality})$, and $\mathsf{Hyb}(\mathsf{Borda}_k, \mathsf{Borda})$ also combine these two properties.

Furthermore, except for STV, all previous hard-to-manipulate protocols involved methods that use pairwise comparisons, and such methods have been criticized for relying too much on the number of victories rather than their magnitude. On the other hand, both $\mathsf{Hyb}(\mathsf{Borda}_k, \mathsf{Plurality})$ and $\mathsf{Hyb}(\mathsf{Borda}_k, \mathsf{Borda})$ are based purely on positional methods, which do not suffer from this flaw, and $\mathsf{Maximin}$ (and hence, hybrids of $\mathsf{Maximin}$ with positional methods) also takes into account the magnitude of victories.

7 Limitations and Extensions

7.1 Hybrids That Are Easy to Manipulate

Unfortunately, our method of obtaining hard-to-manipulate protocols is not universal: if the protocol used in the first phase does not provide the manipulator with sufficiently many choices, the resulting hybrid protocol is almost as easy to manipulate as its second component. In particular, this applies to $\mathsf{Plurality}$ protocol.

Theorem 5. *Suppose that a protocol* Y *satisfies the following property for any candidate* c*: Given other voters' preference profiles, the manipulator can in polynomial time find a beneficial manipulation that ranks* c *first or infer that no such manipulation exists. Then there is a polynomial-time algorithm that can constructively manipulate the hybrid* $\mathsf{Hyb}(\mathsf{Plurality}_k, \mathsf{Y})$ *for any* k.

Proof. For the first phase of the protocol, the only choice that the manipulator has to make is which candidate to rank first; the rest of his vote will have no effect on the elimination process. Hence, he can try all m options. Suppose that when the manipulator ranks c_i first, the set of candidates that survive the first phase is C_i. The manipulator can deduce the honest voters' preferences over C_i. If $c_i \not\in C_i$, he simply has to construct a beneficial manipulation $\pi|_{C_i}$ of Y and, in his vote, rank c_i first and order the candidates in C_i as suggested by $\pi|_{C_i}$. If $c_i \in C_i$, in constructing a beneficial manipulation of Y he is restricted to orderings that rank c_i first. By our assumptions, he can find a solution to this problem in polynomial time. □

Corollary 4. *There are polynomial-time algorithms that can constructively manipulate* $\mathsf{Hyb}(\mathsf{Plurality}_k, \mathsf{Y})$, *where* $\mathsf{Y} \in \{\mathsf{Borda}, \mathsf{Maximin}, \mathsf{BC}, \mathsf{Plurality}\}$ *for any* k.

The property of Plurality that makes it an unsuitable candidate for the first phase of a hybrid protocol is that by altering his vote, the manipulator can obtain at most m different outcomes of the first phase, so he can go over all of them and pick the one that produces best results. It is not clear whether any other protocol for which changing a single vote leads to polynomially many different outcomes is just as bad: each outcome imposes specific restrictions on the manipulator's vote in the second phase, and finding a manipulation that satisfies them may be harder that manipulating the original protocol.

7.2 Other Measures of Complexity

In their paper [4], Conitzer and Sandholm prove that under some pre-round scheduling algorithms, many protocols become #P-hard or PSPACE-hard to manipulate when preceded by a BC pre-round, and [6] shows that one can make manipulation as hard as inverting one-way functions. However, since other protocols that we consider do not have the flexibility provided by the BC scheduling step, the problem of manipulating the hybrids whose first component is not BC, but some other protocol from our list, is inherently in NP. Consequently, a proof that these hybrids are #P-hard or PSPACE-hard to manipulate will lead to a collapse of the polynomial hierarchy, and hence is unlikely.

For the entire class of voting protocols considered in this paper, manipulation is easy when the number of candidates m is very small. This applies both to the standard protocols like STV and to the new hybrid protocols. Indeed, since there are only $m!$ possible ballots for the manipulator, he can go over all of them in order to determine which of them produces the best outcome.

7.3 Utility-Based Voting

In previous sections, we investigated voting schemes that required each voter to submit a total ordering of the candidates. However, in many settings a voter may be essentially indifferent between some of the alternatives, but have a strong opinion on the relative merit of other alternatives. In this case, his preference may be better reflected by a *utility vector* $\mathbf{u} = (u_1, \dots, u_m)$, where $0 \leq u_j \leq 1$ is the *utility* that this voter assigns to candidate c_j. To guarantee fairness, the utility vectors are normalized, i.e., we require that either $u_j = 0$ for all j or $\sum_j u_j = 1$. In addition, we require that all u_j are rational numbers whose representation size is polynomial in n and m.

The definitions of a voting protocol and manipulation can be modified in a straightforward manner. A hybrid of two utility-based protocols is a protocol that performs k steps of the first protocol, re-normalizes the utility vectors (restricted to the surviving candidates) and executes the second protocol on the remaining candidates.

Clearly, a utility vector that assigns different utilities to all candidates can be interpreted as a total ordering of the candidates. Therefore, if all voters rate any two candidates differently, we can interpret their votes as total orderings and use any ordering-based voting protocol. Converse, however, is not true: even if we know all voters' preference orderings, we gain essentially no information as to which candidate maximizes the social welfare (i.e., the sum of all voters' utilities).

More formally, say that a candidate c is *Pareto-dominated* if there is another candidate c' such that all voters rank c below c'. Furthermore, a utility vector is *consistent* with a preference ordering if whenever c_1 is ranked above c_2, the utility assigned to c_1 is strictly larger than the utility assigned to c_2.

Proposition 3. *For any set of preference orderings and any candidate c the following two conditions are equivalent:*

- *c is not Pareto-dominated according to this set;*
- *there is a set of utility vectors consistent with these orderings under which c maximizes the social welfare.*

Proof. Clearly, if c is Pareto-dominated by a c', all voters assign c' a higher utility, so c cannot maximize the sum of voters' utilities. For the other direction, we construct the utility vectors that are consistent with the preference orderings as follows. Given a preference ordering π that has c in the kth position, assign the candidate in the tth position a utility of $\frac{1-\epsilon}{k} + \epsilon\frac{2(m-t)}{m(m-1)}$ if $t \leq k$ and $\epsilon\frac{2(m-t)}{m(m-1)}$ otherwise. Consider any other candidate c'. If a voter ranks c' above c, the utility that she assigns to both alternatives differs by at most ϵ. On the other hand, if she ranks c' below c, then the two utilities differ by at least $\frac{1}{m(m-1)}$. For small enough ϵ, it follows that c has a higher total utility. □

The most natural voting protocol for the utility-based framework is HighestScore, which computes the total score of each candidate, i.e., the sum of utilities assigned to this candidate by all voters, and selects the candidate with the highest total score. However, this protocol is not manipulation-resistant.

Proposition 4. *There is a polynomial-time algorithm that can manipulate* HighestScore.

Proof. Given the utility vectors of all other voters, the manipulator can compute the total scores of all candidates according to other voters' preferences and make a list of candidates whose score differs from the winning score by at most 1 (these are the candidates whom this voter can turn into winners). He can then select his favorite candidate from this list and assign him a utility of 1. Clearly, this is the best outcome that the manipulator can hope to achieve, and, unless the manipulator actually rates all other candidates at 0, this utility vector is not truthful. □

Fortunately, it turns out that the techniques we use for ordering-based protocols are applicable in this setting, too.

A *step* of HighestScore is naturally defined as eliminating the candidate with the lowest score; consequently, the hybrid protocol $\mathsf{Hyb}(\mathsf{HighestScore}_k, \mathsf{HighestScore})$ consists of eliminating k candidates with the lowest score, renormalizing the utility vectors, and choosing the candidate with the highest score among the remaining candidates.

Theorem 6. $\mathsf{Hyb}(\mathsf{HighestScore}_k, \mathsf{HighestScore})$ *is NP-hard to manipulate for infinitely many values of k.*

Proof. To simplify notation, we denote by $u_i(c_j)$ the utility that the voter v_i assigns to the candidate c_j; we omit the index i when it is clear from the context. Whenever $u_i(c_j)$ is not specified, it is assumed to be 0. Also, we denote by $sc(c_j)$ the total score of a candidate c_j.

We present a reduction from EXACT COVER BY 3-SETS. Let ϵ be a small rational number to be specified later. Suppose that an instance of EXACT COVER BY 3-SETS is given by a ground set $G = \{g_1, \ldots, g_N\}$,

$N = 3L$, and a collection $S = \{s_1, \dots, s_M\}$. Let $C^g = \{c^g_1, \dots, c^g_N\}$, $C^s = \{c^s_1, \dots, c^s_M\}$, $C^d = \{c^d_1, \dots, c^d_M\}$, and let the set of candidates be $C = C^g \cup C^s \cup C^d \cup \{p\}$. Suppose that the manipulator's true utilities are $u(p) = 1$, $u(c) = 0$ for $c \neq p$. For each c^g_i, we construct two voters v_i and v'_i whose utilities are $u(c^g_i) = \epsilon$, $u(c^s_j) = (1-\epsilon)/t$ for all j such that $g_i \in s_j$, where $t = |\{s_j \mid g_i \in s_j\}|$, and 13 voters whose utilities are $u(c^g_i) = 1$. Based on these voters' utilities, the scores of all candidates in C^s are between $6(1-\epsilon)/M$ and $6(1-\epsilon)$, so for each candidate c^s_i we additionally construct 6 more voters who distribute their utilities between c^s_i and c^d_i so that when these voters' opinions are taken into account, the scores of all candidates in C^s are exactly equal to 6. Also, for each $c^d_i \in C^d$, we construct 7 voters whose utilities are $u(c^d_i) = 1$. Finally, there are 14 voters whose utilities are $u(p) = 1$, and one voter whose utility is $u(c^g_i) = 1/N$ for all $c^g_i \in C^g$.

We can bound the total scores of all candidates as follows: $sc(p) = 14$, $7 \leq sc(c^d_i) \leq 13$, $sc(c^s_i) = 6$, $sc(c^g_i) = 13 + 2\epsilon + 1/N$.

Set $k = 2L$. The k candidates that will be eliminated in the first phase belong to C^s, and it depends on the manipulator's vote which L of the M candidates in C^s survive the first phase. Suppose that for some c^g_i, all c^s_j such that $g_i \in s_j$ are eliminated. Then after renormalization the utility that v_i and v'_i assign to c^g_i increases from ϵ to 1, and the total score of p increases by at most 1, so we will have $sc(c^g_i) = 15 + 1/N > sc(p)$. On the other hand, if surviving candidates in C^s correspond to a set cover, the score of any candidate in C^g will not exceed $13 + 2M\epsilon + 1/N < 14$ for sufficiently small ϵ, and the scores of all candidates other than p will be less than 14 as well. Therefore, if the manipulator can guess a set cover and assign all corresponding candidates in C^s a utility of $1/L$, he can ensure that p wins, and conversely, if p wins, the k highest-rated candidates in C^s correspond to a set cover. □

Another way to increase resistance to manipulation is to use the method of [4], i.e., prepend HighestScore with a pre-round. A technical difficulty that arises here is that in [4], the pre-round winners are determined on the basis of comparisons, while in our setting, this information may not be available (utility vectors allow for draws). This can be resolved either by requiring the voters to submit an ordering together with their utility vector (clearly, the two should be consistent) or by determining the winner of each pre-round pair by comparing their scores. Both approaches result in hybrid protocols that are NP-hard to manipulate.

8 Conclusions and Future Work

Our work places the results of [3, 4] within a more general paradigm of hybrid voting schemes. The advantage of our approach is that it works for a wide range of protocols: while some voting procedures are inherently hard to manipulate, they may not satisfy the intuitive criteria of a given setting. On the other hand, a hybrid of two protocols retains many of their desirable properties, and sometimes may combine the best of both worlds. All of the voting protocols described in Section 2, as well as many others, are used in different contexts; while it would be unreasonable to expect that all of them will be replaced, say, by STV just because it is harder to manipulate, hybrids of these protocols with similar ones or even with themselves may be eventually preferred to the original protocols. Moreover, our results on utility-based voting suggest that our techniques can be useful for a wider class of problems and can be viewed as a contribution to the more general task of constructing computationally strategy-proof mechanisms.

While we proved that many specific hybrid protocols are hard to manipulate (though some are not), our goal is not to give a complete list of such protocols, or investigate all possible protocol combinations; indeed, given the variety of voting algorithms used in practice, this task seems infeasible. Rather, our work should be viewed as a step towards understanding what makes protocols hard to manipulate, and whether a protocol at hand can be modified to have this property. We believe that the conditions we suggest in our hardness reductions apply in many cases not mentioned in the paper; simplifying these conditions, or replacing them with necessary and sufficient criteria is an interesting open problem.

Another important issue not addressed in this paper is that of designing protocols with high average-case complexity. However, even asking this question properly, i.e., coming up with a natural distribution of voter's preferences with respect to which the average-case hardness is computed is itself a difficult task: clearly, in most scenarios one cannot expect preferences to be uniformly distributed. Initial results in this direction can be found in [6]; however, this topic should be explored further.

Acknowledgements The first author would like to thank Bezalel Gavish for his suggestion to consider utility-based voting systems. Part of this work was done when the first author was visiting Helsinki University of Technology. The authors were partially supported by the Finnish Defence Forces Institute for Technological Research.

References

1. John J. Bartholdi, III, Craig A. Tovey, and James B. Orlin. The Computational Difficulty of Manipulating an Election. *Social Choice and Welfare*, 6:227–241, 1989.
2. John J. Bartholdi, III and James B. Orlin. Single Transferable Vote Resists Strategic Voting. *Social Choice and Welfare*, 8(4):341–354, 1991.
3. Vincent Conitzer and Tuomas Sandholm. Complexity of Manipulating Elections with Few Candidates. In *Proceedings of the Eighteenth National Conference on Artificial Intelligence and Fourteenth Conference on Innovative Applications of Artificial Intelligence*, pages 314–319, Edmonton, Alberta, Canada, July 28 — August 1 2002. AAAI Press.
4. Vincent Conitzer and Tuomas Sandholm. Universal Voting Protocol Tweaks to Make Manipulation Hard. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 781–788, Acapulco, Mexico, August 9–15 2003.
5. Cynthia Dwork, Ravi Kumar, Moni Naor, and D. Sivakumar, Rank aggregation methods for the Web, In *Proc. 10th International World-Wide Web Conference (WWW)*, pages 613–622.
6. Edith Elkind and Helger Lipmaa. Small Coalitions Cannot Manipulate Voting. In *Proceedings of Financial Cryptography and Data Security - Ninth International Conference*, Roseau, The Commonwealth Of Dominica, February 28–March 3, 2005.
7. Eithan Ephrati and Jeffrey S. Rosenschein. Multi-Agent Planning as a Dynamic Search for Social Consensus. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 1993.
8. Allan F. Gibbard. Manipulation of Voting Schemes: A General Result. *Econometrica*, 41:597–601, 1973.
9. Ronald Fagin, Ravi Kumar, Mohammad Mahdian, D. Sivakumar, Erik Vee. Comparing and Aggregating Rankings with Ties. In *Proceedings of 23rd ACM Symposium on Principles of Database Systems (PODS)*, pages 47–58.
10. David M. Pennock, Eric Horvitz, and C. Lee Giles. Social Choice Theory and Recommender Systems: Analysis of the Axiomatic Foundations of Collaborative Filtering. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence*, July 2000.
11. Mark A. Satterthwaite. *The Existence of Strategy-Proof Voting Procedures: A Topic in Social Choice Theory*. PhD thesis, University of Wisconsin, Madison, 1973.