Complexity of unweighted coalitional manipulation under some common voting rules

Lirong Xia, Vincent Conitzer, Ariel D. Procaccia, and Jeffrey S. Rosenschein

Abstract

In this paper, we study the computational complexity of the unweighted coalitional manipulation (UCM) problem under some common voting rules. We show that the UCM problem under maximin is NP-complete. We also show that the UCM problem under ranked pairs is NP-complete, even if there is only one manipulator. Finally, we present a polynomial-time algorithm for the UCM problem under Bucklin.

1 Introduction

Voting is a methodology for a group of agents (or voters) to make a joint choice from a set of alternatives. Each agent reports his or her preferences over the alternatives; then, a *voting rule* is applied to aggregate the preferences of the agents—that is, to select a winning alternative. However, sometimes a subset of the agents can report their preferences insincerely to make the outcome more favorable to them. This phenomenon is known as *manipulation*. A rule for which no group of agents can ever beneficially manipulate is said to be *group strategy-proof*; if no single agent can ever beneficially manipulate, the rule is said to be *strategy-proof* (a weaker requirement).

Unfortunately, any strategy-proof voting rule will fail to satisfy some natural property. The celebrated Gibbard-Satterthwaite theorem [10, 16] states that when there are three or more alternatives, there is no strategy-proof voting rule that satisfies non-imposition (for every alternative, there exist votes that would make that alternative win) and non-dictatorship (the rule does not simply always choose the most-preferred alternative of a single fixed voter). However, the mere existence of beneficial manipulations does not imply that voters will use them: in order to do so, voters must also be able to *discover* the manipulation, and this may be computationally hard. Recently, the approach of using computational complexity to prevent manipulation has attracted more and more attention. In early work [2, 1], it was shown that when the number of alternatives is not bounded, the second-order Copeland and STV rules are hard to manipulate, even by a single voter. More recent research has studied how to modify other existing rules to make them hard to manipulate [3, 7].

Some attention has been given to a problem known as *weighted coalitional manipulation* (WCM) in elections. In this setting, there is a coalition of manipulative voters trying to coordinate their actions in a way that makes a specific alternative win the election. In addition, the voters are weighted; a voter with weight k counts as k voters voting identically. Previous work has established that this problem is computationally hard under a variety of prominent voting rules, even when the number of candidates is constant [6, 11].

However, and quite surprisingly, the current literature contains few results regarding the *unweighted* version of the coalitional manipulation problem (UCM), which is in fact more natural in most settings. Recently, it has been shown that UCM is NP-complete under a family of voting rules derived from the Copeland rule, even with only two manipulators [8]. Zuckerman et al. [20] have established, as corollaries of their main theorems, that unweighted coalitional manipulation is tractable under the Veto and Plurality with Runoff voting rules.

In this paper, we study the computational complexity of the unweighted coalitional manipulation problem under the maximin, ranked pairs, and Bucklin rules. After briefly recalling basic notations and definitions, we show that the UCM problem under maximin is NP-complete for any fixed number of manipulators (at least two). We then show that the UCM problem under ranked pairs is NP-complete, even when there is only one manipulator (just as this is hard for second-order Copeland and STV). Finally, we present a polynomial-time algorithm for the UCM problem under Bucklin.

2 Preliminaries

Let $\mathcal{C}$ be the set of *alternatives* (or *candidates*). A linear order on $\mathcal{C}$ is a transitive, antisymmetric, and total relation on $\mathcal{C}$. The set of all linear orders on $\mathcal{C}$ is denoted by $L(\mathcal{C})$. An n-voter profile P on $\mathcal{C}$ consists of n linear orders on $\mathcal{C}$. That is, $P = (R_1, \ldots, R_n)$, where for every $i \leq n$, $R_i \in L(\mathcal{C})$. The set of all profiles on $\mathcal{C}$ is denoted by $P(\mathcal{C})$. In the remainder of the paper, we let m denote the number of alternatives (that is, $|\mathcal{C}|$).

A *voting rule* r is a function from the set of all profiles on $\mathcal{C}$ to $\mathcal{C}$, that is, $r : P(\mathcal{C}) \to \mathcal{C}$. The following are some common voting rules studied in this paper.

1. *(Positional) scoring rules*: Given a *scoring vector* $\vec{v} = (v(1), \ldots, v(m))$, for any vote $V \in L(\mathcal{C})$ and any $c \in \mathcal{C}$, let $s(V, c) = v(j)$, where j is the rank of c in V. For any profile $P = (V_1, \ldots, V_n)$, let $s(P, c) = \sum_{i=1}^{n} s(V_i, c)$. The rule will select $c \in \mathcal{C}$ so that $s(P, c)$ is maximized. Two examples of scoring rules are *Borda*, for which the scoring vector is $(m-1, m-2, \ldots, 0)$, and *plurality*, for which the scoring vector is $(1, 0, \ldots, 0)$.

2. *Maximin*: Let $N_P(c_i, c_j)$ denote the number of votes that rank c_i ahead of c_j. The winner is the alternative c that maximizes $\min\{N_P(c, c') : c' \in \mathcal{C}, c' \neq c\}$.

3. *Bucklin*: An alternative c's Bucklin score is the smallest number k such that more than half of the votes rank c among the top k alternatives. The winner is the alternative that has the smallest Bucklin score. (Sometimes, ties are broken by the number of votes that rank an alternative among the top k, but for simplicity we will not consider this tie-breaking rule here.)

4. *Ranked pairs* [17]: This rule first creates an entire ranking of all the alternatives. $N_P(c_i, c_j)$ is defined as for the maximin rule. In each step, we consider a pair of alternatives c_i, c_j that we have not previously considered (as a pair): specifically, we choose the remaining pair with the highest $N_P(c_i, c_j)$. We then fix the order $c_i > c_j$, unless this contradicts previous orders that we fixed (that is, it violates transitivity). We continue until we have considered all pairs of alternatives (hence, in the end, we have a full ranking). The alternative at the top of the ranking wins.

All of these rules allow for the possibility that multiple alternatives end up tied for the win. Technically, therefore, they are really *voting correspondences*; a correspondence can select more than one winner. In the remainder of this paper, we will sometimes somewhat inaccurately refer to the above correspondences as rules. We will consider two variants of the manipulation problem: one in which the goal is to make the preferred alternative the unique winner, and one in which the goal is to make sure that the preferred alternative is among the winners. We study the *constructive* manipulation problem, in which the goal is to make a given alternative win.

Definition 1 *An* unweighted coalitional manipulation (UCM) *instance is a tuple* (r, P^{NM}, c, M), *where* r *is a voting rule,* P^{NM} *is the non-manipulators' profile,* c *is the alternative preferred by the manipulators, and* M *is the set of manipulators.*

Definition 2 *The* UCM unique winner (UCMU) *problem is: Given a UCM instance* (r, P^{NM}, c, M), *we are asked whether there exists a profile* P^M *for the manipulators such that* $r(P^{NM} \cup P^M) = \{c\}$.

Definition 3 *The* UCM co-winner (UCMC) *problem is: Given a UCM instance* (r, P^{NM}, c, M), *we are asked whether there exists a profile* P^M *for the manipulators such that* $c \in rP^{NM} \cup P^M$.

3 Maximin

In this section, we show that the UCMU and UCMC problems under maximin are NP-complete, by giving a reduction from the *two vertex disjoint paths in directed graph* problem, which is known to be NP-complete [12].

Definition 4 *The* two vertex disjoint paths in directed graph *problem is: We are given a directed graph G and two disjoint pairs of vertices (u, u') and (v, v'), where u, u', v, v' are all different from each other. We are asked whether there exist two directed paths $u \to u_1 \to \ldots \to u_{k_1} \to u'$ and $v \to v_1 \to \ldots \to v_{k_2} \to v'$ such that $u, u', u_1, \ldots, u_{k_1}, v, v', v_1, \ldots, v_{k_2}$ are all different from each other.*

For any profile P and any pair of alternatives c_1, c_2, let $D_P(c_1, c_2)$ denote the number of times that c_1 is ranked higher than c_2 in P minus the number of times that c_2 is ranked higher than c_1 in P. That is,

$$D_P(c_1, c_2) = |\{R \in P : c_1 \succ_R c_2\}| - |\{R \in P : c_2 \succ_R c_1\}|$$

The next lemma has previously been used by others [13, 4].

Lemma 1 *Given a profile P and $F : \mathcal{C} \times \mathcal{C} \to \mathbb{Z}$ such that*

1. *for all $c_1, c_2 \in \mathcal{C}$, $c_1 \neq c_2$, $F(c_1, c_2) = -F(c_2, c_1)$, and*
2. *either for all pairs of alternatives $c_1, c_2 \in \mathcal{C}$ (with $c_1 \neq c_2$), $F(c_1, c_2)$ is even, or for all pairs of alternatives $c_1, c_2 \in \mathcal{C}$ (with $c_1 \neq c_2$), $F(c_1, c_2)$ is odd,*

there exists a profile P such that for all $c_1, c_2 \in \mathcal{C}$, $c_1 \neq c_2$, $D_P(c_1, c_2) = F(c_1, c_2)$ and $|P| \leq \frac{1}{2} \sum_{c_1, c_2 : c_1 \neq c_2} |F(c_1, c_2) - F(c_2, c_1)|$.

Theorem 1 *The UCMU and UCMC problems under maximin are NP-complete for any fixed number of manipulators (as long as it is at least 2).*

Proof of Theorem 1: It is easy to verify that the UCMU and UCMC problems under maximin are in NP. We first show that UCMU is NP-hard, by giving a reduction from the two vertex disjoint paths in directed graph problem. Let the instance of the two vertex disjoint paths in directed graph problem be denoted by $G = (V, E)$, (u, u') and (v, v') where $V = \{u, u', v, v', c_1, \ldots, c_{m-5}\}$. Without loss of generality, we assume that every vertex is reachable from u or v (otherwise, we can remove the vertex from the instance). We also assume that $(u, v') \notin E$ and $(v, u') \notin E$ (since such edges cannot be used in a solution). Let $G' = (V, E \cup \{(v', u), (u', v)\})$, that is, G' is the graph obtained from G by adding (v', u) and (u', v). We construct a UCMU instance as follows.
Set of alternatives: $\mathcal{C} = \{c, u, u', v, v', c_1, \ldots, c_{m-5}\}$.
Alternative preferred by the manipulators: c.
Number of unweighted manipulators: $|M|$ (for some $|M| \geq 2$).
Non-manipulators' profile: P^{NM} satisfying the following conditions:

1. For any $c' \neq c$, $D_{P^{NM}}(c, c') = -4|M|$.
2. $D_{P^{NM}}(u, v') = D_{P^{NM}}(v, u') = -4|M|$.
3. For any $(s, t) \in E$ such that $D_{P^{NM}}(t, s)$ is not defined above, we let $D_{P^{NM}}(t, s) = -2|M| - 2$.
4. For any $s, t \in \mathcal{C}$ such that $D_{P^{NM}}(t, s)$ is not defined above, we let $|D_{P^{NM}}(t, s)| = 0$.

The existence of such a P^{NM}, whose size is polynomial in m, is guaranteed by Lemma 1.

We can assume without loss of generality that each manipulator ranks c first. Therefore, for any $c' \neq c$, $D_{P^{NM} \cup P^M}(c, c') = -3|M|$.

We are now ready to show that $Maximin(P^{NM} \cup P^M) = \{c\}$ if and only if there exist two vertex disjoint paths from u to u' and from v to v' in G. First, we prove that if there exist such paths in G, then there exists a profile P^M for the manipulators such that $Maximin(P^{NM} \cup P^M) = \{c\}$. Let $u \to u_1 \to \ldots \to u_{k_1} \to u', v \to v_1 \to \ldots \to v_{k_2} \to v'$ be two vertex disjoint paths. Let $V' = \{u, u', v, v', u_1, \ldots, u_{k_1}, v_1, \ldots, v_{k_2}\}$. Then, because any vertex is reachable from u or v in G, there exists a connected subgraph G^* of G' (which still includes all the vertices) in which $u \to u_1 \to \ldots \to u_{k_1} \to u' \to v \to v_1 \to \ldots \to v_{k_2} \to v' \to u$ is the only cycle. Therefore, there exists a linear order O over $V \setminus V'$ such that for any $t \in V \setminus V'$, either 1. there exists $s \in V \setminus V'$ such that $s \succ_O t$ and $(s, t) \in E$, or 2. there exists $s \in V'$ such that $(s, t) \in E$. We let

$$\begin{aligned} P^M = &\{(|M| - 1)(c \succ u \succ u_1 \succ \ldots \succ u_{k_1} \succ u' \succ v \succ v_1 \succ \ldots \succ v_{k_2} \succ v' \succ O)\} \\ &\cup \{c \succ v \succ v_1 \succ \ldots \succ v_{k_2} \succ v' \succ u \succ u_1 \succ \ldots \succ u_{k_1} \succ u' \succ O\} \end{aligned}$$

Then, we have the following calculation

$d_{min} = \min_{c' \neq c} D_{P^{NM} \cup P^M}(c, c') = -4|M| + |M| = -3|M|$.

$D_{P^{NM} \cup P^M}(u, v') = -4|M| + (|M| - 1) - 1 = -3|M| - 2 < -3|M| = d_{min}$.

$D_{P^{NM} \cup P^M}(v, u') = -4|M| + 1 - (|M| + 1) = -5|M| + 2 < -3|M| = d_{min}$.

For any $t \in \mathcal{C} \setminus \{c, u, v\}$, there exists $s \in \mathcal{C} \setminus \{c\}$ such that $(s, t) \in E$ and $D_{P^M}(t, s) = -|M|$, which means that $D_{P^{NM} \cup P^M}(t, s) = -2|M| - 2 - |M| = -3|M| - 2 < -3|M| = d_{min}$.

Hence $Maximin(P^{NM} \cup P^M) = \{c\}$.

Next, we prove that if there exists a profile P^M for the manipulators such that $Maximin(P^{NM} \cup P^M) = \{c\}$, then there exist two vertex disjoint paths from u to u' and from v to v'. We define a function $f : V \to V$ such that $D_{P^{NM} \cup P^M}(t, f(t)) < -3|M|$. We note that since $Maximin(P^{NM} \cup P^M) = \{c\}$, for any $t \neq c$, there must exist s such that $D_{P^{NM} \cup P^M}(t, s) < -3|M|$, and s must be a parent of t in G'. If there exists more than one such s, define $f(t)$ to be any one of them. It follows that if $(t, f(t))$ is neither (u, v') or (v, u'), then $(f(t), t) \in E$ and $D_{P^M}(t, f(t)) = -|M|$, which means that $f(t) \succ t$ in each vote of P^M; otherwise, if $(t, f(t))$ is (u, v') or (v, u'), then $D_{P^M}(t, f(t)) \leq |M| - 2$, which means that $f(t) \succ t$ in at least one vote of P^M. There must exist $l_1 < l_2 \leq m$ such that $f^{l_1}(u) = f^{l_2}(u)$. That is, $f^{l_1}(u), f^{l_1+1}(u), \ldots, f^{l_2-1}(u), f^{l_2}(u)$ is a cycle in G'. We assume that for any $l_1 \leq l_1' < l_2' < l_2$, $f^{l_1'}(u) \neq f^{l_2'}(u)$. Now we claim that (v', u) and (u', v) must be both in the cycle, because

1. if neither of them is in the cycle, then in each vote of P^M, we must have $f^{l_2}(u) \succ f^{l_2-1}(u) \succ f^{l_1}(u) = f^{l_2}(u)$, which contradicts the assumption that each vote is a linear order;

2. if exactly one of them is in the cycle—without loss of generality, $f^{l_1}(u) = v, f^{l_1+1}(u) = u'$—then in at least one of the votes of P^M, we must have $f^{l_2}(u) \succ f^{l_2-1}(u) \succ \ldots \succ f^{l_1}(u) = f^{l_2}(u)$, which contradicts the assumption that each vote is a linear order.

Now, without loss of generality, let us assume that $f^{l_1}(u) = u, f^{l_1+1}(u) = v'$, $f^{l_3}(u) = v, f^{l_3+1}(u) = u'$, where $l_3 \leq l_2 - 2$. We immediately obtain two vertex disjoint paths $u = f^{l_1}(u) = f^{l_2}(u) \to f^{l_2-1}(u) \to \ldots \to f^{l_3+1}(u) = u'$ and $v = f^{l_3}(u) \to f^{l_3-1}(u) \to \ldots \to f^{l_1+1}(u) = v'$. Therefore, UCMU under maximin is NP-complete.

For UCMC, we use almost the same reduction, except we modify it as follows:

2'. Let $D_{P^{NM}}(u, v') = D_{P^{NM}}(v, u') = -4|M| + 2$.

3'. For any $(s, t) \in E$ such that $D_{P^{NM}}(t, s)$ is not defined above, we let $D_{P^{NM}}(t, s) = -2|M|$.

□

4 Ranked pairs

In this section, we prove that the UCMU and UCMC problems under ranked pairs are NP-complete (even for a single manipulator) by giving a reduction from 3SAT.

Definition 5 *The* 3SAT *problem is: Given a set of variables $X = \{x_1, \ldots, x_q\}$ and a formula $Q = Q_1 \wedge \ldots \wedge Q_t$ such that*

1. *for all $1 \leq i \leq t$, $Q_i = l_{i,1} \vee l_{i,2} \vee l_{i,3}$, and*
2. *for all $1 \leq i \leq t$ and $1 \leq j \leq 3$, $l_{i,j}$ is either a variable x_k, or the negation of a variable $\neg x_k$,*

we are asked whether the variables can be set to true or false so that Q is true.

Theorem 2 *The UCMU and UCMC problems under ranked pairs are NP-complete, even when there is only one manipulator.*

Proof of Theorem 2: It is easy to verify that the UCMU and UCMC problems under ranked pairs are in NP. We first prove that UCMU is NP-complete. Given an instance of 3SAT, we construct a UCMU instance as follows. Without loss of generality, we assume that for any variable x, x and $\neg x$ appears in at least one clause, and none of the clauses contain both x and $\neg x$.
Set of alternatives: $\mathcal{C} = \{c, Q_1, \ldots, Q_t, Q'_1, \ldots, Q'_t\} \bigcup \{x_1, \ldots, x_q, \neg x_1, \ldots, \neg x_q\}$ $\bigcup \{Q_{l_{1,1}}, Q_{l_{1,2}}, Q_{l_{1,3}}, \ldots, Q_{l_{t,1}}, Q_{l_{t,2}}, Q_{l_{t,3}}\} \bigcup \{Q_{\neg l_{1,1}}, Q_{\neg l_{1,2}}, Q_{\neg l_{1,3}}, \ldots, Q_{\neg l_{t,1}}, Q_{\neg l_{t,2}}, Q_{\neg l_{t,3}}\}$.
Alternative preferred by the manipulator: c.
Number of unweighted manipulators: $|M| = 1$.
Non-manipulators' profile: P^{NM} satisfying the following conditions.

1. For any $i \leq t$, $D_{P^{NM}}(c, Q_i) = 30$,$D_{P^{NM}}(Q'_i, c) = 20$; for any $x \in \mathcal{C} \setminus \{Q_i, Q'_i : 1 \leq i \leq t\}$, $D_{P^{NM}}(c, x) = 10$.
2. For any $j \leq q$, $D_{P^{NM}}(x_j, \neg x_j) = 20$.
3. For any $i \leq t, j \leq 3$, if $l_{i,j} = x_k$, then $D_{P^{NM}}(Q_i, Q^i_{x_k}) = 30$, $D_{P^{NM}}(Q^i_{x_k}, x_k) = 30$, $D_{P^{NM}}(\neg x_k, Q^i_{\neg x_k}) = 30$, $D_{P^{NM}}(Q^i_{\neg x_k}, Q'_i) = 30$; if $l_{i,j} = \neg x_k$, then $D_{P^{NM}}(Q_i, Q^i_{\neg x_k}) = 30$, $D_{P^{NM}}(Q^i_{x_k}, x_k) = 30$, $D_{P^{NM}}(\neg x_k, Q^i_{\neg x_k}) = 30$, $D_{P^{NM}}(Q^i_{x_k}, Q'_i) = 30$, $D_{P^{NM}}(Q^i_{\neg x_k}, Q^i_{x_k}) = 20$.
4. For any $x, y \in \mathcal{C}$, if $D_{P^{NM}}(x, y)$ is not defined in the above steps, then $D_{P^{NM}}(x, y) = 0$.

For example, when $Q_1 = x_1 \vee \neg x_2 \vee x_3$, $D_{P^{NM}}$ is illustrated in Figure 1.
The existence of such a P^{NM} is guaranteed by Lemma 1, and the size of P^{NM} is in polynomial in t and q.
First, we prove that if there exists an assignment v of truth values to X so that Q is satisfied, then there exists a vote R_M for the manipulator such that $RP(P^{NM} \cup \{R_M\}) = \{c\}$. We construct R_M as follows.

- Let c be on the top of R_M.
- For any $k \leq q$, if $v(x_k) = \top$ (that is, x_k is true), then $x_k \succ_{R_M} \neg x_k$, and for any $i \leq t, j \leq 3$ such that $l_{i,j} = \neg x_k$, let $Q^i_{x_k} \succ_{R_M} Q^i_{\neg x_k}$.
- For any $k \leq q$, if $v(x_k) = \perp$ (that is, x_k is false), then $\neg x_k \succ_{R_M} x_k$, and for any $i \leq t, j \leq 3$ such that $l_{i,j} = \neg x_k$, let $Q^i_{\neg x_k} \succ_{R_M} Q^i_{x_k}$.
- The remaining pairs of alternatives are ranked arbitrarily.

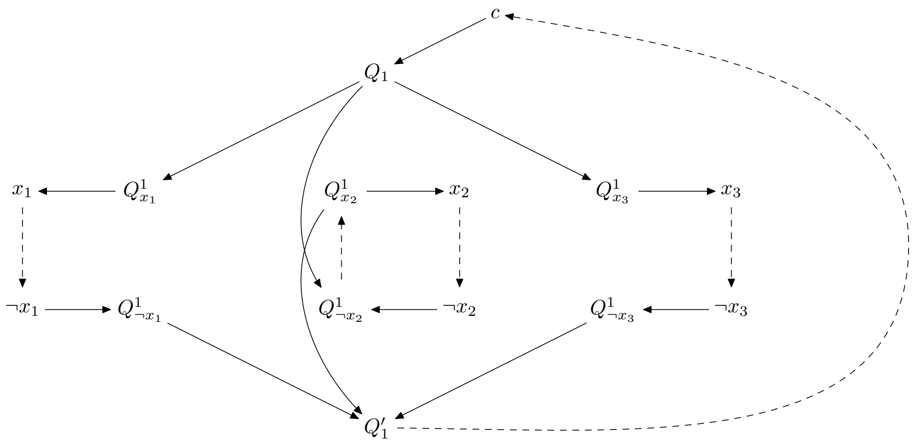

Figure 1: For any vertices v_1, v_2, if there is a solid edge from v_1 to v_2, then $D_{P^{NM}}(v_1, v_2) = 30$; if there is a dashed edge from v_1 to v_2, then $D_{P^{NM}}(v_1, v_2) = 20$; if there is no edge between v_1 and v_2 and $v_1 \neq c$, $v_2 \neq c$, then $D_{P^{NM}}(v_1, v_2) = 0$; for any x such that there is no edge between c and x, $D_{P^{NM}}(c, x) = 10$.

If $x_k = \top$, then $D_{P^{NM} \cup \{R_M\}}(x_k, \neg x_k) = 21$, and for any $i \leq t, j \leq 3$ such that $l_{i,j} = \neg x_k$, $D_{P^{NM} \cup \{R_M\}}(Q^i_{\neg x_k}, Q^i_{x_k}) = 19$. It follows that no matter how ties are broken when applying ranked pairs to $P^{NM} \cup \{R_M\}$, if $x_k = \top$, then $x_k \succ \neg x_k$ in the final ranking. This is because for any $l_{i,j} = \neg x_k$, $D_{P^{NM} \cup \{R_M\}}(Q^i_{\neg x_k}, Q^i_{x_k}) = 19 < 21 = D_{P^{NM} \cup \{R_M\}}(x_k, \neg x_k)$, which means that before trying to fix $x_k \succ \neg x_k$, there is no directed path from $\neg x_k$ to x_k.

Similarly if $x_k = \bot$, then $D_{P^{NM} \cup \{R_M\}}(x_k, \neg x_k) = 19$, and for any $i \leq t, j \leq 3$ such that $l_{i,j} = \neg x_k$, $D_{P^{NM} \cup \{R_M\}}(Q^i_{\neg x_k}, Q^i_{x_k}) = 21$. It follows that if $x_k = \bot$, then $\neg x_k \succ x_k$, and for any $i \leq t, j \leq 3$ such that $l_{i,j} = \neg x_k$, $Q^i_{\neg x_k} \succ Q^i_{x_k}$ in the final ranking. This is because $Q^i_{\neg x_k} \succ Q^i_{x_k}$ will be fixed before $x_k \succ \neg x_k$.

Because Q is satisfied under v, for each clause Q_i, at least one of its three literals is true under v. Without loss of generality, we assume $v(l_{i,1}) = \top$. If $l_{i,1} = x_k$, then before trying to add $Q'_i \succ c$, the directed path $c \rightarrow Q_i \rightarrow Q_{x_k} \rightarrow x_k \rightarrow \neg x_k \rightarrow Q_{\neg x_k} \rightarrow Q'_i$ has already been fixed. Therefore, $c \succ Q'_i$ in the final ranking, which means that for any alternatives x in $\mathcal{C} \setminus \{c, Q_1, \ldots, Q_t, Q'_1, \ldots, Q'_t\}$, $c \succ x$ in the final ranking because $D_{P^{NM} \cup \{R_M\}}(c, x) > 0$. Hence, c is the unique winner of $P^{NM} \cup \{R_M\}$ under ranked pairs.

Next, we prove that if there exists a vote R_M for the manipulator such that $RP(P^{NM} \cup \{R_M\}) = \{c\}$, then there exists an assignment v of truth values to X such that Q is satisfied. We construct the assignment v so that $v(x_k) = \top$ if and only if $x_k \succ_{R_M} \neg x_k$, and $v(x_k) = \bot$ if and only if $\neg x_k \succ_{R_M} x_k$. We claim that $v(Q) = \top$. If, on the contrary, $v(Q) = \bot$, then there exists a clause (Q_1, without loss of generality) such that $v(Q_1) = \bot$. We now construct a way to fix the pairwise rankings such that c is not the winner under ranked pairs, as follows. For any $j \leq 3$, if there exists $k \leq q$ such that $l_{i,j} = \neg x_k$, then $x_k \succ_{R_M} \neg x_k$ because $v(\neg x_k) = \bot$. Therefore, $D_{P^{NM} \cup R_M}(x_k, \neg x_k) = 21$. Then, after trying to add all pairs $x \succ x'$ such that $D_{P^{NM} \cup R_M}(x, x') > 21$ (that is, all solid directed edges in Figure 1), it follows that $x_k \succ \neg x_k$ can be added to the final ranking. We choose to add $x_k \succ \neg x_k$ first, which means that $Q^1_{x_k} \succ Q^1_{\neg x_k}$ in the final ranking (otherwise, we have $Q^1_{\neg x_k} \succ Q^1_{x_k} \succ x_k \succ \neg x_k \succ Q^1_{\neg x_k}$, which is a contradiction).

For any $j \leq 3$, if there exists $k \leq q$ such that $l_{i,j} = x_k$, then $\neg x_k \succ_{R_M} x_k$ because $v(x_k) = \bot$. Therefore, $D_{P^{NM} \cup R_M}(x_k, \neg x_k) = 19$. We note that after trying to add all pairs $x \succ x'$ such that $D_{P^{NM} \cup R_M}(x, x') > 19$, $Q^1_{x_k} \not\succ Q^1_{\neg x_k}$. We recall that for any $j \leq 3$, if there exists $k \leq q$ such that $l_{i,j} = \neg x_k$, then $Q^1_{\neg x_k} \not\succ Q^1_{x_k}$. Hence, it follows that $Q'_1 \succ c$ is consistent with all pairwise rankings added so far. Then, since $D_{P^{NM}\cup R_M}(Q'_1, c) \geq 19$, if $Q'_1 \succ c$ has not been added, we choose to add it first of all pairwise rankings of alternatives $x \succ x'$ such that $D_{P^{NM}\cup R_M}(x, x') = 19$, which means that $Q'_1 \succ c$ in the final ranking—in other words, c is not at the top in the final ranking. Therefore, c is not the unique winner, which contradicts the assumption that $RP(P^{NM} \cup \{R_M\}) = \{c\}$.

For UCMC, we modify the reduction as follows: we let P^{NM} be such that for any $i \leq t$, $D_{P^{NM}}(Q'_i, c) = 22$, and for any $j \leq q$, $D_{P^{NM}}(x_j, \neg x_j) = 22$. □

Similarly, we can prove that when $|M|$ is a constant greater than one, UCMU and UCMC under ranked pairs remain NP-complete.

Theorem 3 *The UCMU and UCMC problems under ranked pairs are NP-complete, even when the number of manipulators is fixed to some constant $|M| > 1$.*

Proof of Theorem 3: We prove UCMU is NP-complete. The proof is similar to that of Theorem 2. We let P^{NM} satisfy the following conditions.

1. For any $i \leq t$, $D_{P^{NM}}(c, Q_i) = 30|M|$,$D_{P^{NM}}(Q'_i, c) = 22|M| - 2$; for any $x \in \mathcal{C} \setminus \{Q_i, Q'_i : 1 \leq i \leq t\}$, $D_{P^{NM}}(c, x) = 10|M|$.

2. For any $j \leq q$, $D_{P^{NM}}(x_j, \neg x_j) = 22|M| - 2$.

3. For any $i \leq t, j \leq 3$, if $l_{i,j} = x_k$, then $D_{P^{NM}}(Q_i, Q^i_{x_k}) = 30|M|$, $D_{P^{NM}}(Q^i_{x_k}, x_k) = 30|M|$, $D_{P^{NM}}(\neg x_k, Q^i_{\neg x_k}) = 30|M|$, $D_{P^{NM}}(Q^i_{\neg x_k}, Q'_i) = 30|M|$; if $l_{i,j} = \neg x_k$, then $D_{P^{NM}}(Q_i, Q^i_{\neg x_k}) = 30|M|$, $D_{P^{NM}}(Q^i_{x_k}, x_k) = 30|M|$, $D_{P^{NM}}(\neg x_k, Q^i_{\neg x_k}) = 30|M|$, $D_{P^{NM}}(Q^i_{x_k}, Q'_i) = 30|M|$, $D_{P^{NM}}(Q^i_{\neg x_k}, Q^i_{x_k}) = 20|M|$.

4. For any $x, y \in \mathcal{C}$, if $D_{P^{NM}}(x, y)$ is not defined in the above steps, then $D_{P^{NM}}(x, y) = 0$.

First, if there exists an assignment v of truth values to X so that Q is satisfied, then we let R_M be defined as in the proof for Theorem 2. It follows that $RP(P^{NM} \cup \{|M|R_M\}) = \{c\}$ (all the manipulators can vote R_M).

Next, if there exists a profile P^M for the manipulators such that $RP(P^{NM} \cup P^M) = \{c\}$, then we construct the assignment v so that $v(x_k) = \top$ if $x_k \succ_V \neg x_k$ for all $V \in P^M$, and $v(x_k) = \bot$ if $\neg x_k \succ_V x_k$ for all $V \in P^M$; the values of all the other variables are assigned arbitrarily. Then by similar reasoning as in the proof for Theorem 2, we know that Q is satisfied under v.

For UCMC, the proof is similar (by slightly modifying the $D_{P^{NM}}$ as we did in the proof of Theorem 2). □

5 Bucklin

In this section, we present a polynomial-time algorithm for the UCMU problem under Bucklin (a polynomial-time algorithm for the UCMC problem under Bucklin can be obtained similarly). For any alternative x, any natural number d, and any profile P, let $B(x, d, P)$ denote the number of times that x is ranked among the top d alternatives in P. The idea behind the algorithm is as follows. Let d_{min} be the minimal depth so that c is ranked among the top d_{min} alternatives in more than half of the votes (when all of the manipulators rank c first). Then, we check if there is a way to assign the manipulators' votes so that none of the other alternatives is ranked among the top d_{min} alternatives in more than half of the votes.

Algorithm 1
Input: *A UCM instance $(Bucklin, P^{NM}, c, M)$, $C = \{c, c_1, \ldots, c_{m-1}\}$.*

1. *Calculate the minimal depth d_{min} such that $B(c, d_{min}, P^{NM}) + |M| > \frac{1}{2}(|NM| + |M|)$.*

2. *If there exists $c' \in C$, $c' \neq c$ such that $B(c', d_{min}, P^{NM}) > \frac{1}{2}(|NM| + |M|)$, then output that there is no successful manipulation. Otherwise, for any $c' \in C$, $c' \neq c$, let $d_{c'} = \lfloor \frac{1}{2}(|NM| + |M|) \rfloor - B(c', d_{min}, P^{NM})$, $k_{c'} = \begin{cases} |M| & \text{if } d_{c'} \geq |M| \\ d_{c'} & \text{otherwise} \end{cases}$.*

3. *If $\sum_{c' \neq c} k_{c'} < (d_{min} - 1)|M|$, then output that there is no successful manipulation.*

4. *Let $j = 1$, $t = 1$, and for any $l \leq |M|$, let R_l rank c at the top. Repeat Step 4a $m - 1$ times:*

4a. *If $k_{c_t} > 0$, then c_t is ranked in the next position (lower than the candidates that have already been ranked in previous steps) in $R_{\bmod (j-1,|M|)+1}, R_{\bmod (j,|M|)+1}, \ldots, R_{\bmod (j+k_{c_t}-2,|M|)+1}$, respectively, where for any natural number a, b, $\bmod (a, b)$ is the common residue of $a (\bmod b)$. Let $j \leftarrow \bmod (j + k_{c_t} - 1, |M|) + 1$, $t \leftarrow t + 1$.*

5. *For any $s \leq |M|$, complete R_s arbitrarily. Output $P^M = (R_1, \ldots, R_{|M|})$.*

Claim 1 *Algorithm 1 correctly solves the UCMU problem. It runs in time $O(m|NM| + |NM||M| + |M|m)$.*

Proof of Claim 1: Let us first consider the case where Algorithm 1 outputs that there is no successful manipulation. There are two cases.

1. There exists $c' \in C$, $c' \neq c$ such that $B(c', d_{min}, P^{NM}) > \frac{1}{2}(|NM| + |M|)$.
2. $\sum_{c' \neq c} k_{c'} < (d_{min} - 1)|M|$. In this case, for any P^M, there exists $c' \neq c$ such that $|M| \geq B(c', d_{min}, P^M) > k_{c'}$, which means that $B(c', d_{min}, P^{NM} \cup P^M) > \frac{1}{2}(|NM| + |M|)$.

In both cases, more than half of the voters rank c' among the top d_{min} alternatives. Therefore, c cannot be the unique winner.

Now let us consider the case where Algorithm 1 outputs some P^M. In this case, for any $t \leq m - 1$, $B(c_t, d_{min}, P^M) \leq k_{c_t}$. Therefore, for any $t \leq m - 1$, $B(c_t, d_{min}, P^{NM} \cup P^M) \leq B(c_t, d_{min}, P^{NM}) + k_{c_t} \leq \frac{1}{2}(|NM| + |M|)$, which means that $Bucklin(P^{NM} \cup P^M) = \{c\}$.

Step 1 runs in time $O(m|NM|)$, Step 2 runs in time $O(|M||NM|)$, Step 3 runs in time $O(|M|)$, and Step 4 and Step 5 run in time $O(m|M|)$. Therefore, Algorithm 1 runs in time $O(m|NM| + |NM||M| + |M|m)$. $\Box$

6 Discussion

Number of manipulators	1	constant
Copeland (specific tie-breaking)	P [2]	NP-hard [8]
STV	NP-hard [1]	NP-hard [1]
Veto	P [20]	P [20]
Plurality with Runoff	P [20]	P [20]
Cup	P [6]	P [6]
Maximin	P [2]	**NP-hard**
Ranked pairs	**NP-hard**	**NP-hard**
Bucklin	**P**	**P**
Borda	P [2]	?

Table 1: Complexity of UCM under prominent voting rules. Boldface results appear in this paper.

In this paper, we studied the computational complexity of unweighted coalitional manipulation under the maximin, ranked pairs, and Bucklin rules. The UCM problem is NP-complete under the maximin rule for any fixed number (at least two) of manipulators. The UCM problem is also NP-complete under the ranked pairs rule; in this case, the hardness holds even if there is only a single manipulator, similarly to the second-order Copeland and STV rules. We gave a polynomial-time algorithm for the UCM problem under the Bucklin rule. Table 1 summarizes our results, and puts them in the context of previous results on the UCM problem.

It should be noted that all of these hardness results, as well as the ones mentioned in the introduction, are *worst-case* results. Hence, there may still be an efficient algorithm that can find a beneficial manipulation for *most* instances. Indeed, several recent results suggest that finding manipulations is usually easy. Procaccia and Rosenschein have shown that, when the number of alternatives is a constant, manipulation of positional scoring rules is easy even with respect to "junta" distributions, which arguably focus on hard instances [15]. Conitzer and Sandholm have given some sufficient conditions under which manipulation is easy and argue that these conditions are usually satisfied in practice [5]. Zuckerman et al. have given manipulation algorithms with the property that if they fail to find a manipulation when one exists, then, if the manipulators are given some additional vote weights, the algorithm will succeed [20]. The asymptotic probability of manipulability has also been characterized (except for knife-edge cases) for a very general class of voting rules [18] (building on earlier work [14]). In a similar spirit, several quantitative versions of the Gibbard-Satterthwaite theorem have recently been proved [9, 19]. One weakness of all of these results (except [20]) is that they make assumptions about the distribution of instances. In this paper, we have focused on the worst-case framework, which does not suffer from this weakness. This does mean that when we show that manipulation is hard, it may still be the case that it is usually easy.

There are many interesting problems left for future research. For example, settling the complexity of UCM under positional scoring rules such as Borda is a challenging open problem.

Acknowledgments

We thank anonymous reviewers for helpful comments and suggestions. This work is supported in part by the United States-Israel Binational Science Foundation under grant 2006-216. Lirong Xia is supported by a James B. Duke Fellowship, Vincent Conitzer is supported by an Alfred P. Sloan Research Fellowship, and Ariel Procaccia is supported by the Adams Fellowship Program of the Israel Academy of Sciences and Humanities. Xia and Conitzer are also supported by the NSF under award number IIS-0812113.

References

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

[2] John Bartholdi, III, Craig Tovey, and Michael Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241, 1989.

[3] Vincent Conitzer and Tuomas Sandholm. Universal voting protocol tweaks to make manipulation hard. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 781–788, Acapulco, Mexico, 2003.

[4] Vincent Conitzer and Tuomas Sandholm. Common voting rules as maximum likelihood estimators. In *Proceedings of the 21st Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 145–152, Edinburgh, UK, 2005.

[5] Vincent Conitzer and Tuomas Sandholm. Nonexistence of voting rules that are usually hard to manipulate. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, Boston, MA, 2006.

[6] Vincent Conitzer, Tuomas Sandholm, and Jérôme Lang. When are elections with few candidates hard to manipulate? *Journal of the ACM*, 54(3):Article 14, 1–33, 2007. Early versions in AAAI-02 and TARK-03.

[7] Edith Elkind and Helger Lipmaa. Hybrid voting protocols and hardness of manipulation. In *Annual International Symposium on Algorithms and Computation (ISAAC)*, 2005.

[8] Piotr Faliszewski, Edith Hemaspaandra, and Henning Schnoor. Copeland voting: Ties matter. In *To appear in Proceedings of AAMAS'08*, 2008.

[9] Ehud Friedgut, Gil Kalai, and Noam Nisan. Elections can be manipulated often. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, 2008.

[10] Allan Gibbard. Manipulation of voting schemes: a general result. *Econometrica*, 41:587–602, 1973.

[11] Edith Hemaspaandra and Lane A. Hemaspaandra. Dichotomy for voting systems. *Journal of Computer and System Sciences*, 73(1):73–83, 2007.

[12] Andrea S. LaPaugh and Ronald L. Rivest. The subgraph homeomorphism problem. In *Proceedings of the tenth annual ACM symposium on Theory of computing (STOC)*, pages 40–50, 1978.

[13] David C. McGarvey. A theorem on the construction of voting paradoxes. *Econometrica*, 21(4):608–610, 1953.

[14] Ariel D. Procaccia and Jeffrey S. Rosenschein. Average-case tractability of manipulation in voting via the fraction of manipulators. In *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, Honolulu, HI, USA, 2007.

[15] Ariel D. Procaccia and Jeffrey S. Rosenschein. Junta distributions and the average-case complexity of manipulating elections. *Journal of Artificial Intelligence Research (JAIR)*, 28:157–181, 2007.

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

[17] T. N. Tideman. Independence of clones as a criterion for voting rules. *Social Choice and Welfare*, 4(3):185–206, 1987.

[18] Lirong Xia and Vincent Conitzer. Generalized scoring rules and the frequency of coalitional manipulability. In *Proceedings of the Ninth ACM Conference on Electronic Commerce (EC)*, 2008.

[19] Lirong Xia and Vincent Conitzer. A sufficient condition for voting rules to be frequently manipulable. In *Proceedings of the Ninth ACM Conference on Electronic Commerce (EC)*, 2008.

[20] Michael Zuckerman, Ariel D. Procaccia, and Jeffrey S. Rosenschein. Algorithms for the coalitional manipulation problem. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2008.

Lirong Xia
Department of Computer Science
Duke University
Durham, NC 27708, USA
Email: `lxia@cs.duke.edu`

Vincent Conitzer
Department of Computer Science
Duke University
Durham, NC 27708, USA
Email: `conitzer@cs.duke.edu`

Ariel Procaccia
School of Engineering and Computer Science
The Hebrew University of Jerusalem
Jerusalem, Israel
Email: `arielpro@cs.huji.ac.il`

Jeffrey S. Rosenschein
School of Engineering and Computer Science
The Hebrew University of Jerusalem
Jerusalem, Israel
Email: `jeff@cs.huji.ac.il`