On the Computational Hardness of Manipulating Pairwise Voting Rules

Rohit Vaish
Indian Institute of Science
India
rohit.vaish@csa.iisc.ernet.in

Neeldhara Misra
IIT Gandhinagar
India
mail@neeldhara.com

Shivani Agarwal
Indian Institute of Science
India
shivani@csa.iisc.ernet.in

Avrim Blum
Carnegie Mellon University
USA
avrim@cs.cmu.edu

ABSTRACT

Standard voting rules usually assume that the preferences of voters are provided in the form of complete rankings over a fixed set of alternatives. This assumption does not hold in applications like recommendation systems where the set of alternatives is extremely large and only partial preferences can be elicited. In this paper, we study the problem of strategic manipulation of voting rules that aggregate voter preferences provided in the form of pairwise comparisons between alternatives. Our contributions are twofold: first, we show that any onto pairwise voting rule is manipulable in principle. Next, we analyze how the computational complexity of manipulation of such rules varies with the structure of the graph induced by the pairs of alternatives that the manipulator is allowed to vote over and the type of the preference relation. Building on natural connections between the pairwise manipulation and sports elimination problems (including a mixed-elimination variant that we introduce in this paper), we show that manipulating pairwise voting rules can be computationally hard even in the single-manipulator setting, a setting where most standard voting rules are known to be easy to manipulate.

General Terms

Algorithms, Economics, Theory

Keywords

Social Choice Theory; Voting; Manipulation; Pairwise Preferences

1. INTRODUCTION

A central result in social choice theory [1] is the Gibbard-Satterthwaite theorem [2, 3] which states that any non-dictatorial voting rule over at least three candidates under which each candidate has some chance of winning is susceptible to strategic voting i.e. is *manipulable*. Given the impossibility suggested by this theorem, there has been substantial work concerning a finer analysis of the situation and finding possible workarounds. A prominent example of this is the seminal work of Bartholdi, Tovey and Trick [4] who proposed the possibility of using computational hardness as a barrier against manipulation of voting rules. They argue that although opportunities for manipulation always exist in principle, there might not exist any efficient general purpose algorithm for finding them in practice.

Since then, a considerable body of work has developed around the computational study of manipulation (see [5] for a survey on this topic). Much of this work models voter preferences as *complete rankings* over a fixed set of alternatives (or candidates). While this is a reasonable choice for certain situations, many large-scale settings prevalent today (like recommendation systems) involve extremely large candidate sets (such as movies, products, webpages etc.). It is therefore unreasonable, and often impractical, to elicit preferences from users in the form of complete rankings of these alternatives. For these settings, it is decidedly more natural to aggregate *partial* user preferences to arrive at an outcome (e.g. top-k preferences [6, 7], partial orders [8] etc.). Similarly, some situations call for relaxing the requirement of *transitivity* among the preferences. Indeed, when alternatives are compared using not one but multiple quality parameters, it is natural to permit possibly cyclic preferences [9].

In this work, we consider the model of *pairwise preferences*, where every vote is simply a collection of pairwise comparisons between alternatives with no constraints other than anti-symmetry (i.e. $A \succ B \Rightarrow B \not\succ A$). The preferences provided by voters in this setting can be *incomplete* (i.e. not all pairs of candidates are compared by a voter) and can contain *cycles* ($A \succ B, B \succ C, C \succ A$). Thus, all the situations described above are subsumed by this model. This is a simple model that offers substantial generality, and as a result there has been growing interest in designing preference aggregation algorithms that elicit pairwise preferences from the users [10, 11, 12, 13, 14]. Although these studies explore the statistical properties of aggregation algorithms for pairwise preferences, the question of whether these algorithms (or *pairwise voting rules*) are resistant to strategic user behavior remains to be answered. The focus of the present work is to address these questions from both axiomatic and computational perspectives.

Appears in: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016), J. Thangarajah, K. Tuyls, C. Jonker, S. Marsella (eds.), May 9–13, 2016, Singapore.*

$\mathcal{A}$	pref-type: strict + acyclic	acyclic	strict	unrestricted
tree/ max deg=2	**P**	**P**	**P**	**P**
complete	**P**	**NP-complete**	**NP-complete**	**NP-complete**
bipartite	**P**	**NP-complete**	**NP-complete**	**NP-complete**
general	**P**	**NP-complete**	**NP-complete**	**NP-complete**

Table 1: *The complexity of* pBorda-Manipulation *under the conditions specified by the corresponding action space $\mathcal{A}$ and type of preference relation* `pref-type` *(refer Definition 2.1 in Section 2).*

Contributions

Our contributions can be summarized as follows:

1. We show that any pairwise voting rule that is onto (i.e. under which each candidate has some chance of winning) must be manipulable (Theorem 3.1).

2. Given the above impossibility, we study settings under which computational difficulty acts as a worst-case barrier against manipulation (Sections 3.2 and 3.3) Specifically, we study the problem of manipulation of pairwise voting rules along the following two dimensions: (i) the *structure of the graph* induced by the set of candidate pairs that the manipulator is allowed to vote over (e.g. tree, bipartite, complete) and (ii) the *type of preference relation* (e.g. strict, acyclic). Our analysis is focused on the *pairwise Borda* (pBorda) rule and the Copeland$^\alpha$ family of voting rules. Tables 1 to 4 summarize our results for these rules. The essence of our results is that *the structure of the induced graph and the type of revealed preferences (and the parameter α in case of* Copeland$^\alpha$*) can shape the complexity landscape in important ways.* We remark that while the manipulation problem in the context of rankings is polynomially solvable for most voting rules [4], we already encounter hardness results in the setting of pairwise comparisons with one manipulator for the relatively simple pBorda and Copeland$^\alpha$ rules, which we consider an important contrast[1].

Organization of this paper.

After setting the requisite notation and definitions in Section 2, we describe the main ideas and proof techniques involved in the axiomatic (Section 3.1) and classical complexity-theoretic results for pairwise Borda (Section 3.2) and Copeland$^\alpha$ (Section 3.3) rules. A survey of the related literature is provided in Section 4. We conclude by identifying further implications of our results and directions for future research in Section 5. Due to space constraints, we defer the detailed proofs of all the results presented here to the full version of the paper.

[1]Even in the context of rankings, however, voting rules like the second-order Copeland rule [4] and many elimination-style rules [15, 16, 17, 18] are known to be computationally resistant to manipulation by a single voter.

$\mathcal{A}$	pref-type: strict + acyclic	acyclic	strict	unrestricted
tree/ max deg=2	**P**	**P**	**P**	**P**
complete	**P**	**P**	**P**	**P**
bipartite	**P**	**P**	**P**	**P**
general	**P**	**P**	**P**	**P**

Table 2: *The complexity of* Copeland^0-Manipulation *and* Copeland^1-Manipulation.

$\mathcal{A}$	pref-type: strict + acyclic	acyclic	strict	unrstd.
tree/ max deg=2	**P**	**P**	**P**	**P**
complete	**P**	**P**	**NP-complete**	**P**
bipartite	**P**	**P**	**NP-complete**	**P**
general	**P**	**P**	**NP-complete**	**P**

Table 3: *The complexity of* Copeland$^{0.5}$-Manipulation.

$\mathcal{A}$	pref-type: strict + acyclic	acyclic	strict	unrestricted
tree/ max deg=2	**P**	**P**	**P**	**P**
complete	**P**	**NP-complete**	**NP-complete**	**NP-complete**
bipartite	**P**	**NP-complete**	**NP-complete**	**NP-complete**
general	**P**	**NP-complete**	**NP-complete**	**NP-complete**

Table 4: *The complexity of* Copeland$^\alpha$-Manipulation *for $\alpha \in \mathbb{Q} \cap (0,1) \setminus \{0.5\}$.*

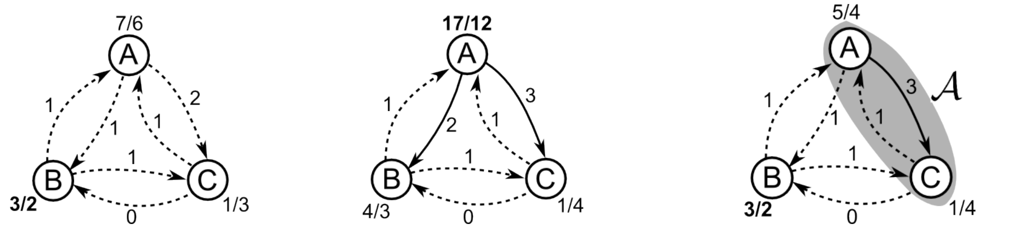
2. PRELIMINARIES

Let $[n] = \{1, 2, \ldots, n\}$ denote the set of *candidates* and $\mathcal{U} = \{u_1, u_2, \ldots, u_m\}$ denote the set of *voters* in an election.

Pairwise preferences and pairwise voting rules.

Let $\succ_u \subseteq [n] \times [n]$ denote the binary relation indicating the preferences of voter u, so that $i \succ_u j$ indicates that voter u prefers candidate i over candidate j. For each pair of candidates i, j and each voter u, we can have exactly one of $i \succ_u j$, $j \succ_u i$ or neither (i.e. voter u *skips* the comparison between i and j). We let $\mathcal{R}$ denote the set of all such anti-symmetric and irreflexive binary relations on $[n]$; and let $\Pi = (\succ_{u_1}, \succ_{u_2}, \cdots, \succ_{u_m}) \in \mathcal{R}^m$ denote the *pairwise preference profile* of the voters.

A *pairwise voting rule* r maps a pairwise preference profile $\Pi \in \cup_{k=1}^{\infty} \mathcal{R}^k$ to a unique candidate $r(\Pi) \in [n]$. Given a preference profile $\Pi \in \mathcal{R}^m$ and a pair of candidates i, j, let $m_{ij}(\Pi)$ denote the number of voters who strictly prefer candidate i over candidate j, i.e. $m_{ij}(\Pi) = \sum_{k=1}^{m} \mathbf{1}(i \succ_{u_k} j)$ where $\mathbf{1}(.)$ is the indicator function. A *score-based pairwise voting rule* is any pairwise voting rule r for which there exists a (natural) scoring function $\mathbf{s} : \cup_{k=1}^{\infty} \mathcal{R}^k \to \mathbb{R}^n$ such that $r(\Pi)$ is the highest-scoring candidate according to $\mathbf{s}(\Pi)$ under some fixed tie-breaking rule. That is, $r(\Pi) = T(\arg\max_i s_i(\Pi))$ for some tie-breaking rule $T : 2^{[n]} \setminus \{\emptyset\} \to [n]$ satisfying $T(S) \in S$ for



(a) *Before manipulation* **(b)** *After manipulation* **(c)** *After restricted manipulation*

Figure 1: *An illustration of the election instance in Example 2.1. (a) Each vertex of the multigraph represents a candidate and each dashed edge represents the number of voters with that preference (e.g. two voters prefer $A \succ C$). (b) The pairwise comparisons made by the manipulator are represented by solid edges and the* pBorda *score of the winning candidate is indicated in boldface. (c) The restricted action space of the manipulator ($\mathcal{A} = \{(A, C)\}$) is shaded in grey.*

all non-empty $S \subseteq [n]$. Some examples of score-based pairwise voting rules are as follows:

(i) *Pairwise Borda Rule* (pBorda) [14]: The pBorda score of candidate i under preference profile Π is given by[2]:

$$s_i^{\text{pBorda}}(\Pi) = \sum_{j=1}^{n} \frac{m_{ij}(\Pi)}{m_{ij}(\Pi) + m_{ji}(\Pi)}.$$

(ii) *Copeland*$^\alpha$ *Rule* [19]: The Copeland$^\alpha$score ($\alpha \in [0, 1]$) of candidate i under preference profile Π is given by:

$$s_i^{\text{Copeland}^\alpha}(\Pi) = \sum_{j=1}^{n} \mathbf{1}\big(m_{ij}(\Pi) > m_{ji}(\Pi)\big) + \alpha \cdot \mathbf{1}\big(m_{ij}(\Pi) = m_{ji}(\Pi)\big).$$

Manipulation of pairwise voting rules.

We focus on manipulation by a single strategic voter (the *manipulator*) who has *complete information* about the votes of all other voters (the *non-manipulators*). Formally, a pairwise voting rule r is said to be *manipulable* if there exists a pair of profiles $\Pi = (\succ_{u_1}, \succ_{u_2}, \ldots, \succ_{u_m}), \Pi' = (\succ_{u_1}, \succ_{u_2}, \ldots, \succ_{u_{m-1}}, \succ'_{u_m}) \in \mathcal{R}^m$ differing only in the preference of voter u_m such that $r(\Pi') \succ_{u_m} r(\Pi)$. That is, the manipulator u_m strictly prefers the new outcome over the old one. The corresponding computational problem, referred to as r-Manipulation, is defined as follows:

Definition 2.1. r-Manipulation

Instance: *A tuple $\langle \Pi, i^*, \mathcal{A}, \texttt{pref-type} \rangle$ where $\Pi \in \mathcal{R}^{m-1}$ is the preference profile of the non-manipulators $(u_1, u_2, \ldots, u_{m-1})$, $i^* \in [n]$ is the distinguished candidate, $\mathcal{A} \subseteq \binom{[n]}{2}$ is the set of pairwise comparisons that the manipulator is allowed to make and $\texttt{pref-type} \in$ {strict+acyclic, strict, acyclic, unrestricted} is the preference constraint with respect to $\mathcal{A}$.*

Question: *Does there exist a vote $\succ_{u_m}$ over $\mathcal{A}$ satisfying $\texttt{pref-type}$ such that $r((\Pi, \succ_{u_m})) = i^*$?*

Here $\mathcal{A} \subseteq \binom{[n]}{2}$ denotes the *action space* of the manipulator i.e. the pairs of candidates that the manipulator is allowed to vote over. Alternately, no pair of candidates outside $\mathcal{A}$ can be compared by the manipulator. Given $\mathcal{A}$, we say that a vote $\succ_u$ by a voter u over $\mathcal{A}$ has a *strict* preference type (i.e. $\texttt{pref-type}$ = strict) if for all candidate pairs $\{i, j\} \in \mathcal{A}$, either $i \succ_u j$ or $j \succ_u i$ (i.e. voter u is not allowed to skip comparisons). Similarly, a vote satisfying $\texttt{pref-type}$ = acyclic is not allowed to contain directed cycles (i.e. $1 \succ_u 2, 2 \succ_u 3, 3 \succ_u 1$ is forbidden etc.). A vote with $\texttt{pref-type}$ = strict+acyclic is required to simultaneously satisfy the strictness and acyclicity constraints while $\texttt{pref-type}$ = unrestricted imposes none of these restrictions.

We will refer to the instantiation of r-Manipulation for pBorda rule (resp. Copeland$^\alpha$) as pBorda-Manipulation (resp. Copeland$^\alpha$-Manipulation). Example 2.1 illustrates the role of the space $\mathcal{A}$ in the manipulation problem.

Example 2.1 (The role of action space $\mathcal{A}$). *Consider the election setting shown in Figure 1a, where the* pBorda *scores of the candidates A, B & C respectively are* 7/6, 3/2 *&* 1/3 *and B is the* pBorda *winner. Suppose we now add the manipulator u_4 to this election whose favorite candidate is A. Observe that if the manipulator casts the vote $\{(A \succ B), (A \succ C)\}$ (see Figure 1b), the new* pBorda *scores for A, B & C will be* 17/12, 4/3 *&* 1/4 *respectively and A becomes the winner. Thus, the answer to* pBorda-Manipulation *for this election instance is YES when $\mathcal{A} = \{(A, B), (A, C)\}$ or $\mathcal{A} = \{(A, B), (A, C), (B, C)\}$ and* $\texttt{pref-type}$ = unrestricted. *However, if the manipulator is only allowed to compare the candidates A and C, then despite voting in favor of A, the manipulator cannot make A win (Figure 1c). Therefore, the answer to* pBorda-Manipulation *is NO when $\mathcal{A} = \{(A, C)\}$. It is easy to see that the above observations also hold for* $\texttt{pref-type} \in$ {strict, acyclic, strict+acyclic}.

Excess scores.

The excess score of a candidate i is the amount by which the score of i exceeds the score of the distinguished candidate i^* in a given election. For instance, in Figure 1c, the excess pBorda scores of candidates B and C (with respect to distinguished candidate A) are $1/4$ and -1 respectively. Hence, r-Manipulation for a score-based voting rule r can be restated as finding a vote for the manipulator such that the final excess scores of all candidates are zero or less.

Elimination problem in sports.

The elimination problem [20] asks whether a team i^* can still win a sports competition, given the current scores and set of remaining games. As we will see, this problem turns out to be intimately connected to r-Manipulation.

[2] where we adopt the convention $0/0 = 0$.

Formally, let $[N] = \{1, 2, \ldots, N\}$ be the set of N teams and let $s_i \in \mathbb{R}$ denote the current score of team $i \in [N]$. Let $\mathcal{G} \subseteq \binom{[N]}{2}$ denote the set of remaining games between pairs of teams . A *scoring system* S determines the points awarded to each team in a pairwise game based on the outcome of the game (e.g. *home win*, *draw* or *away win*).

DEFINITION 2.2. SCORING SYSTEM: *A scoring system* S *is a tuple of ordered pairs of rational numbers* $[(\alpha_1, \beta_1), (\alpha_2, \beta_2), \cdots, (\alpha_t, \beta_t)]$ *where the subscript* $1, 2, \cdots, t$ *corresponds to the outcome of a game and the first and second entries of each pair (denoted by* α *and* β*) correspond to the points awarded respectively to the home and away team under that outcome. For example, the well-known European football scoring system, where* $\mathrm{S}_1 = [(3, 0), (1, 1), (0, 3)]$ *and* $t = 3$*, awards* 3 *points for win,* 1 *point for draw and* 0 *for loss, regardless of home-away distinction. Similarly,* $\mathrm{S}_2 = [(3, 0), (1, 2), (0, 3)]$ *awards an extra point to the away team under a draw outcome. Finally,* $\mathrm{S}_3 = [(3, 0), (0, 3)]$ *allows only win-loss outcomes and the winner gets* 3 *points.*

Given the current scores $\mathbf{s}$ and the set of games to be played between the teams $\mathcal{G}$, S-ELIMINATION asks the question "*can team* i^* *still win?*" [21]

DEFINITION 2.3. S-ELIMINATION

Instance: *A tuple* $\langle \mathbf{s}, i^*, \mathcal{G} \rangle$ *where* $\mathbf{s} = (s_1, s_2, \ldots, s_N)^T$ *is the vector of current scores of the* N *teams,* $i^* \in [N]$ *is the distinguished team and* $\mathcal{G} \subseteq \binom{[N]}{2}$ *is the set of remaining games between the teams.*

Question: *Does there exist an assignment of outcomes for the games in* $\mathcal{G}$ *such that* i^* *ends up with the (joint) highest total score among all teams under the scoring system* S *?*

A comprehensive characterization of the computational difficulty of this question was provided by Kern and Paulusma [21] in terms of the nature of scoring system S used to award points to the teams.

THEOREM 2.1 (KERN AND PAULUSMA [21]). *Let* S *be a scoring system satisfying*[3] $\alpha_1 > \cdots > \alpha_{t-1} = 1 > \alpha_t = 0$ *and* $0 = \beta_1 < 1 \leq \beta_2 < \cdots < \beta_t$*. Then* S-ELIMINATION *is polynomially solvable if* $\mathrm{S} = \{(t-1-i, i) : 0 \leq i \leq t-1\}$ *for some* $t \in \mathbb{N}$*. In all other cases, the problem is* NP-complete.

3-dimensional Matching [22].

DEFINITION 2.4. 3-D MATCHING

Instance: *A tuple* $\langle W, X, Y, R \rangle$ *where* W, X, Y *are disjoint sets each of size* q *and* $R \subseteq W \times X \times Y$ *is a set of triples such that each* $w \in W$ *(similarly* $x \in X$ *and* $y \in Y$*) features in exactly two triples.*

Question: *Does there exist a subset* $R' \subseteq R$ *such that* R' *covers* $W \cup X \cup Y$ *exactly i.e. each* $w \in W$ *(similarly* $x \in X, y \in Y$*) is present in exactly one triple in* R'*?*

3-D MATCHING remains NP-complete even when each element of W, X, Y occurs in exactly two triples in R [23].

[3] This is without loss of generality, since any given instance of S-ELIMINATION can be transformed into an equivalent instance where the scoring system satisfies these conditions by the normalization procedure described in [21].

3. OUR RESULTS AND TECHNIQUES

3.1 Axiomatic result

In its standard form, the Gibbard-Satterthwaite theorem [2, 3] requires voter preferences over the set of candidates to be complete, transitive and unrestricted (i.e. voters are allowed to pick any possible ranking). Several follow-up works have studied generalizations of this result to settings such as (i) minimally rich subsets of the domain of rankings [24, 25, 26, 27], (ii) incomplete preferences such as partial orders or top-k choices [7, 8], (iii) voting rules with multiple winners [28, 8], etc. Our axiomatic result can be seen as belonging to this line of research where we simultaneously relax the assumptions of transitivity and completeness.

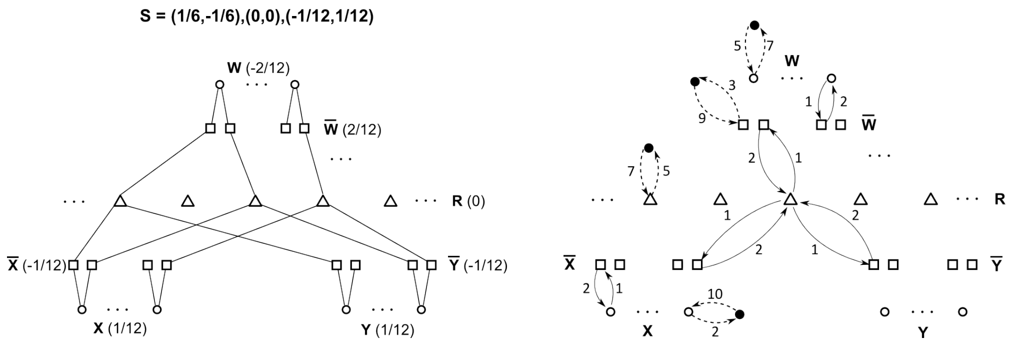
Specifically, we show that the inevitability of strategic voting extends to the more general domain of pairwise preferences. We might expect this intuitively – indeed, if r is a non-manipulable pairwise voting rule, then the projection of r to the domain of rankings will also be non-manipulable. However, the projection of a "non-dictatorial" pairwise voting rule r to the domain of rankings might still turn out to be dictatorial, making the desired contradiction elusive.

THEOREM 3.1 (AXIOMATIC RESULT). *If there are at least three candidates and at least two voters, then any pairwise voting rule that is onto must also be manipulable.*

PROOF. Let us assume that the voting rule r is onto. Consider a preference profile Π_1 where all votes are identical directed cycles of the form $n \succ n-1$, $n-1 \succ n-2$, $\cdots$, $2 \succ 1$, $1 \succ n$. We require $n \geq 3$ for this to be well defined. Let $r(\Pi_1) = 1$. Starting from voter u_1, sequentially modify the votes of all voters $u_1, u_2, \ldots, u_m$ such that in each vote, 2 beats all other candidates and 1 beats everyone except for 2. At each stage of this modification process, 1 must remain the election winner (if 2 wins at any stage, then the swing voter has an incentive to switch to the new vote; if some other candidate $i \neq 1, 2$ wins, then the swing voter can switch back to the old vote). Call this new profile Π_2. Hence $r(\Pi_2) = 1$. Ontoness of r implies that there exists a preference profile Π such that $r(\Pi) = 2$. Without loss of generality, Π can be transformed into a profile Π_3 where 2 beats everyone in each vote and continues to be the winner, i.e. $r(\Pi_3) = 2$. Starting from Π_3, sequentially modify the votes to transform it into Π_2. At some point during this process 2 must lose, providing the desired manipulation. □

3.2 Complexity of manipulating pairwise Borda

We build a comprehensive landscape of the computational complexity of pBorda-MANIPULATION for various combinations of possibilities that arise along the two dimensions mentioned earlier, namely (i) the structure of the *action space* ($\mathcal{A}$ = tree, bipartite, complete graph etc.) and (ii) the *preference type* (`pref-type` = strict, acyclic etc.). Our results show that the manipulation problem turns out to be easy (i.e. polynomial time) whenever the graph structure is *simple enough* (e.g. tree/max degree=2), regardless of the preference type. However, for more complex structures like bipartite/complete graphs, requiring either strictness or acyclicity (but not both) can lead to computational hardness (i.e. NP-hardness). Our results for pBorda-MANIPULATION are summarized in Table 1 and stated as Theorems 3.2 to 3.5.



(a) *Sports instance* **(b)** *Election instance*

Figure 2: *Figure (a) on the left shows the graph structure of the reduced* S-Elimination *instance in [21] with the scores of the teams listed inside parentheses. Figure (b) on the right shows the graph structure of the corresponding* pBorda-Manipulation *instance. The solid circles in figure (b) correspond to the dummy candidates and the dashed/solid edges represent the votes of the non-manipulators for comparisons involving/not involving a dummy candidate.*

Theorem 3.2. pBorda-Manipulation *is polynomially solvable in the following cases: (a) for any choice of* `pref-type` *when $\mathcal{A}$ is a* tree *and (b) for any choice of $\mathcal{A}$ when* `pref-type` = strict+acyclic.

Proof. *(Sketch.)* Part (a) above is shown by a bottom-up greedy algorithm that starts at the leaves of the tree and forces an option for the manipulator without loss of generality. For example, if a leaf has a positive excess score, then it must lose against its parent (and if this is not enough, then it is legitimate to abort). Similarly, if a leaf has negative excess, then it might as well win the pairwise comparison with its parent whenever the score transfer is feasible.

Part (b) is shown by a greedy approach similar to the one used in the classical setting of rankings [4]. At each step, the algorithm checks whether some vertex can be made a *source* vertex for the remaining graph (i.e. all its incident edges are oriented as outgoing). If yes, the algorithm updates the excess scores for all vertices and proceeds with the remaining (smaller) graph; otherwise, the algorithm outputs No. □

Remark 3.1. *The proofs for both parts (a) and (b) of Theorem 3.2 rely only on a locality property of score-based pairwise voting rules — which is that adding a pairwise comparison between a pair of candidates only affects the scores of the two candidates involved. For this reason, other score-based pairwise voting rules with this locality property (e.g.* Copeland$^\alpha$*) also admit polynomial time manipulation algorithms in the settings described above (refer Tables 2 to 4).*

Theorem 3.3. pBorda-Manipulation *is* NP-complete *for $\mathcal{A}$ ∈* {bipartite, general graph} *and* `pref-type` ∈ {acyclic, unrestricted}.

Proof. We reduce from the instance of S-Elimination with S $= [(1/6, -1/6), (0, 0), (-1/12, 1/12)]$ that was used by Kern and Paulusma [21] in their NP-hardness proof [4]. Specifically, the instance consists of seven groups of teams comprising $[N]$, namely $W, \bar{W}, R, \bar{X}, \bar{Y}, X$ and Y, represented as circles, squares and triangles in Figure 2a. The current scores of all teams in each group are listed inside parentheses. Also as shown, each team in the groups W, X, Y has two remaining games against teams in $\bar{W}, \bar{X}, \bar{Y}$ resp., while each team in the group R has three remaining games - one apiece against teams in $\bar{W}, \bar{X}, \bar{Y}$. The distinguished team i^* (not shown in the figure) has no remaining games and has an initial score of 0. Finally, for any game in $\mathcal{G}$ the team from the groups $\bar{W}, \bar{X}, \bar{Y}$ should be considered the *away* team.

The reduced election instance (Figure 2b) is constructed as follows: the set of candidates $[n]$ consists of (i) a candidate i for each team $i \in [N]$ (including a candidate i^* for the team i^*) and (ii) the dummy candidates $N+1, \ldots, N+\ell$ (shown as solid circles) where $\ell \leq N$ i.e. at most one per team (hence $n = N + \ell$). The action space $\mathcal{A}$ of the manipulator corresponds to the set of remaining games $\mathcal{G}$. The votes of non-manipulators (twelve in total — $u_1, \ldots, u_{12}$) are set up as follows: (i) *votes between candidate pairs (i, j) in $\mathcal{A}$*: for each game $(i, j) \in \mathcal{G}$ where i is the home team and j is the away team, the votes between the corresponding candidate pair $(i, j) \in \mathcal{A}$ are set up in a 1:2 configuration[5]; (ii) *votes involving dummy candidates*: for each candidate corresponding to a team in the groups $W, \bar{W}, R, X$ or Y, we add a dummy candidate and set the votes between them in the configurations 7:5, 3:9, 5:7, 2:10 or 2:10 respectively (see Figure 2b); and (iii) *votes involving the distinguished candidate i^**: the pairs $(i^*, N+1)$ and $(i^*, N+2)$ are set up in the configurations 1:0 and 5:7 respectively.

It is easy to check that the excess scores of the candidates match the excess scores of the corresponding teams in the sports instance. Besides, for each candidate pair in $\mathcal{A}$ (setup in a 1:2 configuration), the pBorda scores of the pair can change by $(1/6, -1/6)$,$(0, 0)$ or $(-1/12, 1/12)$ depending on how the manipulator votes, which is exactly the scoring system S in original sports instance. The equivalence of the two solutions is now straightforward.

Note that the graph $\mathcal{G}$ in the original sports instance (Figure 2a) is *bipartite*. Besides, any valid orientation of the remaining games in that construction turns out to be

[4]The proof in [21] was actually presented for S-Elimination with S $= [(3, 0), (1, 2), (0, 3)]$, which can be reduced to our problem using the following normalization scheme: start by awarding, for each remaining game in $\mathcal{G}$, 1 and 2 points respectively to each home and away team *in advance*, followed by scaling down all score values by 12.

[5]A *vote configuration* of 1:2 between a pair of candidates (i, j) means that one voter (say u_1) votes $i \succ_{u_1} j$ while two other voters (say u_2, u_3) vote $j \succ_{u_2} i$ and $j \succ_{u_3} i$.

acyclic. As a consequence, pBorda-Manipulation remains NP-complete when $\mathcal{A}$ = bipartite graph and `pref-type` = acyclic. The implications for $\mathcal{A}$ = general graph and `pref-type` = unrestricted follow. □

Remark 3.2. *Since the above reduction requires at most* 12 *non-manipulators,* pBorda-Manipulation *is NP-complete* even with $\mathcal{O}(1)$ non-manipulators.

Our next result (Theorem 3.4) shows that pBorda-Manipulation is NP-complete when the manipulator is not allowed to skip any pairwise comparison in $\mathcal{A}$ (i.e. `pref-type` = strict). Our proof consists of two parts: first, we introduce a generalization of S-Elimination (which we call Mixed-Elimination) and show that the problem is NP-complete via reduction from 3-D Matching (Lemma 3.1). Next, we show that Mixed-Elimination reduces to pBorda-Manipulation when `pref-type` = strict. We start by describing the problem of Mixed-Elimination.

Mixed-Elimination generalizes S-Elimination to competitions where different games can be played under different scoring systems. That is, instead of a single scoring system S as in S-Elimination, Mixed-Elimination consists of k different scoring systems $\mathrm{S}_1, \mathrm{S}_2, \dots, \mathrm{S}_k$ and an assignment function η which maps each game in $\mathcal{G}$ to exactly one of these scoring systems i.e. $\eta(\mathcal{G}) \in [k]^{|\mathcal{G}|}$. As before, the question of interest is: *can team i^* still win?*

Definition 3.1. Mixed-Elimination$(\mathrm{S}_1, \mathrm{S}_2, \dots, \mathrm{S}_k)$

Instance: *A tuple $\langle \mathbf{s}, i^*, \mathcal{G}, \eta \rangle$ where $\mathbf{s} = (s_1, s_2, \dots, s_N)^T$ is the vector of current scores of the N teams, $i^* \in [N]$ is the distinguished team, $\mathcal{G} \subseteq \binom{[N]}{2}$ is the set of remaining games between the teams and η is a function that assigns a scoring system from $\{\mathrm{S}_1, \mathrm{S}_2, \dots, \mathrm{S}_k\}$ to each game in $\mathcal{G}$.*

Question: *Does there exist an assignment of outcomes for the games in $\mathcal{G}$ which when scored according to the scoring systems chosen by η make i^* the team with the (joint) highest score?*

Our proof of Theorem 3.4 focuses on Mixed-Elimination$(\mathrm{S}_1, \mathrm{S}_2)$ where $\mathrm{S}_1 = [(10, -10), (-5, 5)]$ and $\mathrm{S}_2 = [(8, -8), (-2, 2)]$. Thus, each remaining game is scored according to either S_1 or S_2 and no game can end in a draw. We already know from Theorem 2.1 that the *pure* elimination problems with either of these scoring systems (i.e. S_1-Elimination or S_2-Elimination) admit polynomial time algorithms[6]. As we show below, however, the *mixed* elimination problem with these two scoring systems together is computationally hard.

Lemma 3.1. Mixed-Elimination$(\mathrm{S}_1, \mathrm{S}_2)$ *with* $\mathrm{S}_1 = [(10, -10), (-5, 5)]$ *and* $\mathrm{S}_2 = [(8, -8), (-2, 2)]$ *is* NP-complete *when the set of remaining games $\mathcal{G}$ induces a bipartite graph.*

Proof. We reduce from a variant of 3-D Matching where each element of the sets W, X, Y occurs in exactly two triples in R [23] (refer Definition 2.4). Our proof follows the template of a similar proof in [21] where a reduction from the standard 3-D Matching problem is used to show hardness of S-Elimination. The main idea of the proof in [21] is to treat the nodes in the *matching graph* of the given 3-D Matching instance (see Figure 3) as teams and the edges as the set of remaining games between teams. We use this idea to set up a sports instance where the games in the upper half of the matching graph are scored according to $\mathrm{S}_1 = [(10, -10), (-5, 5)]$ and those in lower half according to $\mathrm{S}_2 = [(8, -8), (-2, 2)]$, shown as solid and dashed edges in Figure 3.

Formally, given an instance $\langle W, X, Y, R \rangle$ of 3-D Matching, we construct an instance $\langle \mathbf{s}, i^*, \mathcal{G}, \eta \rangle$ of Mixed-Elimination$(\mathrm{S}_1, \mathrm{S}_2)$ as follows: (i) the *set of teams* $[N]$ is the union of the sets $W, X, Y, \bar{W}, \bar{X}, \bar{Y}, R$ (i.e. one team per element) along with the distinguished team i^*; (ii) the *set of remaining games* $\mathcal{G}$ is precisely the set of all solid and dashed edges in the matching graph; (iii) the *assignment function* η is such that all games involving teams from $\bar{W}$ are scored according to S_1, while all other games in $\mathcal{G}$ are scored according to S_2. Furthermore, for any game, a team *higher up* in the matching graph should be considered the *away* team; and (iv) the *current score* of the team i^* is fixed at 0 without loss of generality. The excess scores of all teams in the sets $W, \bar{W}, R, \bar{X}, \bar{Y}, X$ and Y are respectively $0, -1, 0, -1, -1, -7$ and -7. We now show the equivalence of the two instances.

($\Rightarrow$) Suppose there exists a 3-dimensional matching $R' \subseteq R$. Then a winning assignment can be constructed as follows: (i) each team $r = (w, x, y)$ in the matching set (i.e. $r \in R'$) beats the team $\bar{w} \in \bar{W}$ corresponding to the element w, and loses to the teams $\bar{x} \in \bar{X}$ and $\bar{y} \in \bar{X}$ corresponding to the element x (and y); (ii) each $r \in R \setminus R'$ does the exact opposite by losing to the team above it and beating the teams below it in the matching graph and (iii) if a team $\bar{w} \in \bar{W}$ loses the game against the team in R, then it wins its only other remaining game against w and vice versa. Similarly for teams in $\bar{X}$ and $\bar{Y}$. It can now be easily checked that the above assignment of outcomes is a winning one for team i^*.

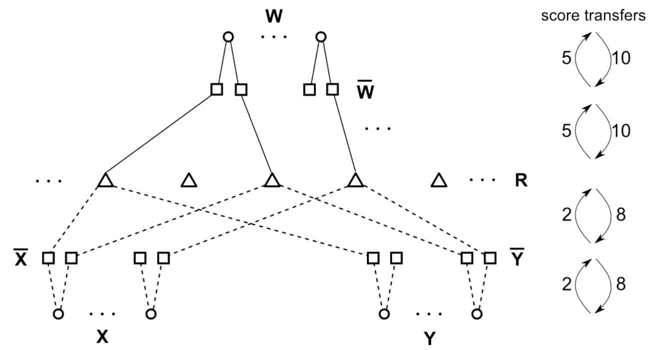


Figure 3: *This figure shows the matching graph for a given* 3-D Matching *instance (left) and score transfers between teams at each level (right). The base sets W, X, Y and the set of triples R are represented as circles and triangles respectively. Each element of the set W (resp. X, Y) is connected to two elements of the set $\bar{W}$ (resp. $\bar{X}, \bar{Y}$) (shown as square nodes) representing the two occurrences of this element in R.*

[6]This is because any instance of S-Elimination with $\mathrm{S} = \mathrm{S}_1$ can be transformed into an equivalent (efficiently solvable) instance of S-Elimination with $\mathrm{S} = [(1, 0), (0, 1)]$ by first, awarding *in advance* -5 and -10 points respectively to each home and away team for each remaining game in $\mathcal{G}$; followed by scaling down all team scores by 15. Similarly for $\mathrm{S} = \mathrm{S}_2$.

(⇐) Now suppose there exists an assignment of the remaining games that makes team i^* win. Without loss of generality, each team in R under such an assignment either wins its game against the team in $\bar{W}$ (in which case it loses against teams in $\bar{X}$ and $\bar{Y}$) or loses against the team in $\bar{W}$ (in which case it wins against the teams in $\bar{X}$ and $\bar{Y}$). The set of teams in R that win against teams in $\bar{W}$ can now be shown to constitute a valid 3-dimensional matching.

Finally, observe that the graph $\mathcal{G}$ in the reduced instance is bipartite. This proves Lemma 3.1. □

THEOREM 3.4. pBorda-MANIPULATION *is* NP-complete *for* $\mathcal{A} \in$ {bipartite, general graph} *and* `pref-type` = strict.

PROOF. *(Sketch.)* Our proof uses a reduction from MIXED-ELIMINATION(S_1, S_2) with $S_1 = [(10, -10), (-5, 5)]$ and $S_2 = [(8, -8), (-2, 2)]$, which was shown to be NP-complete in Lemma 3.1 even when the graph induced by the set of remaining games is bipartite. Just like in the proof of Theorem 3.3, we match teams with candidates and the set of remaining games with the action space of the manipulator. Furthermore, if a game between two teams is scored according to S_1 (respectively S_2), then the votes of non-manipulators between the corresponding candidates are set up in a 1:2 (respectively 1:4) configuration. The win-loss-only condition translates to `pref-type` = strict. □

REMARK 3.3. *It is worth pointing out that when* `pref-type`=strict*, we cannot show hardness for* pBorda-MANIPULATION *by reduction from* S-ELIMINATION*. The reason is that for a scoring system S to correspond to the changes in* pBorda *scores due to the manipulator's vote, it must be of the form* $[(\alpha, -\alpha), (0, 0), (-\beta, \beta)]$*, where* $\alpha, \beta \in \mathbb{N}$*. When skipping a comparison is not allowed (as is the case with* `pref-type`=strict*), the required form becomes* $S = [(\alpha, -\alpha), (-\beta, \beta)]$*, which can in turn be reduced (via translation and scaling) to the trivial* $[(0, 0)]$ *system. This motivates the need to consider instances with more than one scoring system, as in* MIXED-ELIMINATION.

THEOREM 3.5. pBorda-MANIPULATION *is* NP-complete *for* $\mathcal{A}$ = complete graph *and* `pref-type` ∈ {strict, acyclic, unrestricted}.

PROOF. *(Sketch.)* For any fixed choice of `pref-type`, we reduce from the corresponding pBorda-MANIPULATION problem with $\mathcal{A}$ = general graph, which was shown to be NP-complete in Theorems 3.3 and 3.4. The set of candidates in the original and the reduced instances are identical. The votes of non-manipulators between candidate pairs within $\mathcal{A}$ in the original instance are also identical to the votes between corresponding candidate pairs in the reduced instance. For candidate pairs outside $\mathcal{A}$ in the original instance that are set up in $a : b$ configuration, the corresponding candidate pairs in the reduced instance are set up in $aK : bK$ configuration, where K is sufficiently large (yet polynomial in n, m). 'Scaling up' votes in this manner not only preserves the exact pBorda scores (and therefore the excess scores) of the candidates in the reduced instance, but it also forces the manipulator to effectively work with candidate pairs corresponding to the action space A of the original instance. This is because a large number of non-manipulators' votes between a pair of candidates nullifies the change in pBorda scores brought about by the manipulator's vote. □

3.3 Complexity of manipulating Copeland$^\alpha$

Our results for the complexity of Copeland$^\alpha$-MANIPULATION are summarized in Tables 2 to 4 and are stated as Theorems 3.6 to 3.8.

The family of Copeland$^\alpha$ voting rules (parameterized by α) was first studied by Faliszewski et al. [19] in the context of manipulation by a coalition of two voters (recall from Remark 3.1 that manipulation by a single voter is polynomial time). Together with [29], these papers showed that Copeland$^\alpha$-MANIPULATION with two manipulators is NP-complete for all $\alpha \in \mathbb{Q} \cap [0, 1] \setminus \{0.5\}$ when $\mathcal{A}$ = complete graph and `pref-type` = strict+acyclic (i.e. when votes are complete rankings). However, when votes are allowed to be *tournaments* (i.e. $\mathcal{A}$ = complete graph, `pref-type` = strict), Copeland$^\alpha$-MANIPULATION was shown to admit a polynomial time algorithm for $\alpha \in \{0, 1\}$ for any given number ($k \geq 1$) of manipulators[7] [29] . This algorithm can be easily modified to work for any general graph $\mathcal{A}$. Our first result in this section (Theorem 3.6) generalizes the tractability of Copeland$^\alpha$-MANIPULATION for $\alpha \in \{0, 1\}$ to any choice of $\mathcal{A}$ when `pref-type` ∈ {strict, acyclic, unrestricted} (Table 2). We also show that Copeland$^{0.5}$-MANIPULATION is efficiently solvable for any choice of $\mathcal{A}$ when `pref-type` ∈ {acyclic, unrestricted} (Table 3).

THEOREM 3.6. Copeland$^\alpha$-MANIPULATION *is polynomially solvable in the following cases: (a) for* $\alpha \in \{0, 1\}$ *for any choice of* $\mathcal{A}$ *when* `pref-type` ∈ {strict, acyclic, unrestricted} *and (b) for* $\alpha = 0.5$ *for any choice of* $\mathcal{A}$ *when* `pref-type` ∈ {acyclic, unrestricted}.

PROOF. *(Sketch.)* Since the manipulator cannot affect the Copeland outcomes for pairwise contests between candidate pairs (i, j) with $|m_{ij} - m_{ji}| > 1$, we assume without loss of generality that the manipulator either *skips* such comparisons (if `pref-type` ∈ {acyclic, unrestricted}) or provides *arbitrary preferences* over them (if `pref-type` = strict). Hence, for the rest of the proof, we will only consider the pairs (i, j) where $|m_{ij} - m_{ji}| \leq 1$.

Part (a): We start with Copeland^0-MANIPULATION. Here, the manipulator prefers to have as many *ties* between pairs of non-distinguished candidates as possible. Thus, when `pref-type` = unrestricted, the manipulator votes $j \succ i$ whenever $m_{ij} = m_{ji} + 1$ and skips the comparison otherwise.

When `pref-type` = strict, the manipulator can no more skip the comparisons with $m_{ij} = m_{ji}$. For such pairs, the Copeland scores of candidates (i, j) can change by $(+1, 0)$ or $(0, +1)$ for the vote $i \succ j$ or $j \succ i$ of the manipulator. This subproblem is identical to S-ELIMINATION with $S = [(1, 0), (0, 1)]$, which is efficiently solvable (Theorem 2.1).

Finally, when `pref-type` = acyclic, the manipulator once again votes $j \succ i$ whenever $m_{ij} = m_{ji} + 1$ and skips the comparison otherwise. Such a vote suffices if already acyclic.

[7]Theorem 5.2 in [29] states that Copeland$^{0.5}$-MANIPULATION (given $k \geq 1$ manipulators) is also in P when $\mathcal{A}$ = complete graph and `pref-type` = strict. The proof refers to a related problem called *microbribery* of Copeland$^\alpha$ elections, which was shown to be tractable only for $\alpha \in \{0, 1\}$ but not for $\alpha = 0.5$ [30]. We fix this by showing that Copeland$^{0.5}$-MANIPULATION is in fact NP-complete when $\mathcal{A} \in$ {bipartite, complete graph} and `pref-type` = strict (Theorem 3.8), implying the same for the corresponding microbribery problem (of which the manipulation problem is a special case).

Otherwise, we claim that it suffices to run the greedy algorithm from Theorem 3.2 over the action space $\mathcal{A}$ restricted to the pairs with $m_{ij} = m_{ji} + 1$ (and skip all other comparisons). Indeed, notice that the Copeland scores of such pairs (i, j) can change by $(0, 0)$, $(0, 0)$ or $(-1, 0)$ for the comparisons $i \succ j$, skip or $j \succ i$ respectively. Any valid vote that skips such a comparison can therefore be replaced by a vote that makes a strict comparison instead. Thus, the restriction of any valid vote to such pairs has `pref-type` = strict+acyclic for which the greedy algorithm from Theorem 3.2 suffices.

Now consider Copeland^1-MANIPULATION. Here, the manipulator tries to *avoid ties* between non-distinguished candidates by mimicing the majority vote for pairs (i, j) where $m_{ij} = m_{ji} + 1$ and either skipping (if `pref-type` = unrestricted) or using the same S-ELIMINATION routine as above for the remaining pairs (if `pref-type` = strict). The case of `pref-type` = acyclic is also handled similarly.

Part (b): For Copeland$^{0.5}$-MANIPULATION and `pref-type` = unrestricted, the Copeland scores of a pair (i, j) with $m_{ij} = m_{ji} + 1$ can change by $(0, 0)$, $(0, 0)$ or $(-0.5, +0.5)$ for the comparisons $i \succ j$, skip or $j \succ i$ respectively. Similarly, for a pair (i, j) with $m_{ij} = m_{ji}$, the change can be $(+0.5, -0.5)$, $(0, 0)$ or $(-0.5, +0.5)$ respectively. Using the normalization scheme suggested in [21], this problem can be shown to be an instance of MIXED-ELIMINATION$(\mathrm{S}_1, \mathrm{S}_2)$ with $\mathrm{S}_1 = [(1, 0), (0, 1)]$ and $\mathrm{S}_2 = [(2, 0), (1, 1), (0, 2)]$, which is efficiently solvable using the maximum flow formulation of [21].

When `pref-type` = acyclic, the vote constructed above suffices if already acyclic. Otherwise, any directed cycle can be completely replaced by an equivalent undirected cycle (i.e. skip votes). This is because an incoming and an outgoing edge for a vertex together amount to a simultaneous decrease and increase of 0.5, which is the same as 'no change in score' due to the undirected edges. □

Our next result shows that Copeland$^\alpha$-MANIPULATION is NP-complete for all $\alpha \in \mathbb{Q} \cap (0, 1) \setminus \{0.5\}$ when $\mathcal{A} \in$ {bipartite, complete graph} and `pref-type` $\in$ {acyclic, unrestricted} (Table 4). The reduction is from S-ELIMINATION with $\mathrm{S} = [(1/\alpha, 0), (1, 1), (0, 1/\alpha)]$ which is NP-complete for all scoring systems except for $\alpha = 0.5$ (Theorem 2.1). We omit the proof due to space constraints and note that it closely follows the proof of Theorem 3.3.

THEOREM 3.7. Copeland$^\alpha$-MANIPULATION *is* NP-complete *for all* $\alpha \in \mathbb{Q} \cap (0, 1) \setminus \{0.5\}$ *when* $\mathcal{A} \in$ {bipartite, complete graph} *and* `pref-type` $\in$ {acyclic, unrestricted}.

Our final result shows that Copeland$^\alpha$-MANIPULATION is NP-complete for all $\alpha \in \mathbb{Q} \cap (0, 1)$ when the manipulator is required to provide strict preferences over a bipartite/complete graph. The reduction is from MIXED-ELIMINATION$(\mathrm{S}_1, \mathrm{S}_2)$ with $\mathrm{S}_1 = [(1, 0), (0, 1)]$ and $\mathrm{S}_2 = [(1 - \alpha, 0), (0, \alpha)]$, which in turn can be shown to be NP-complete for all $\alpha \in \mathbb{Q} \cap (0, 1)$ via reduction from 3-D MATCHING following the template outlined in the proof of Theorem 3.4. We omit the proof due to space constraints.

THEOREM 3.8. Copeland$^\alpha$-MANIPULATION *is* NP-complete *for all* $\alpha \in \mathbb{Q} \cap (0, 1)$ *when* $\mathcal{A} \in$ {bipartite, complete graph} *and* `pref-type` = strict.

4. RELATED WORK

Manipulation under partial preferences: A number of recent studies have looked at the problem of manipulation of voting rules when voters provide partial preferences in the form of top-k choices or rankings with ties [31, 32, 33, 34]. However, most voting rules used in these studies crucially depend on the *position* of a candidate in each vote, making it difficult to generalize them to the pairwise setting. An exception is the Copeland$^\alpha$ family of rules, for which the coalitional manipulation problem has been studied in the context of weighted votes. Even here, a direct comparison with our setting is not possible because these papers study a more general problem than ours (namely, coalitional manipulation with weighted votes) over a strict subdomain of pairwise preferences (namely, partial rankings).

Possible and Necessary Winners: By far, the most common approach for dealing with incomplete preferences in the computational social choice literature is via *possible* and *necessary winners* [35, 36, 37, 38, 39, 40, 41]. A *possible* (respectively *necessary*) winner for an incomplete preference profile is a candidate chosen by the voting rule for at least one (respectively all) completion(s) of the partial votes. The notion of a completion (or extension) is crucial to these studies because the voting rules considered by them are well-defined only when either all votes are provided as complete rankings or when all pairs of candidates have been compared by each voter. We, on the other hand, consider pairwise voting rules (e.g. pBorda, Copeland$^\alpha$) that deal with incomplete preferences *directly* without the need for extending each incomplete vote.

5. CONCLUDING REMARKS

We studied the problem of manipulation of voting rules in the model of pairwise preferences and showed results of both axiomatic and computational flavor. In particular, we provided a complete understanding of how the computational complexity of manipulating pairwise Borda (pBorda) rule and Copeland$^\alpha$ family of voting rules is influenced by the action space of the manipulator and the type of preference relation, up to the point of distinguishing the polynomial time cases from the NP-complete ones.

Our results extend quite naturally to many other closely related problems like *destructive manipulation* (i.e. making a distinguished candidate lose) or generalizations such as *top-k manipulation*, *bottom-k manipulation* (i.e. ensuring a top-k or bottom-k finish for a distinguished candidate) etc. Further, many of our results rely only on a certain *locality property* of the pairwise voting rule and turn out to be true for any pairwise voting rule that has this property.

The most natural direction for future research would be to understand the complexity of manipulation in the pairwise preference model for other voting rules like PageRank [42], HodgeRank [11], Ranked Pairs, Schulze's rule [43] etc. An ambitious question in this context would be a complete classification of the complexity of manipulation (by a single voter) in the space of voting rules on pairwise preferences. Another direction involves studying the parameterized complexity landscape of the manipulation problem. Our existing results already have some implications, but several other cases are open.

6. ACKNOWLEDGEMENTS

This work was supported in part by NSF grant CCF-1525971 and a DST INSPIRE Faculty grant [IFA12-ENG-31].

REFERENCES

[1] Kenneth J Arrow, Amartya Sen, and Kotaro Suzumura. *Handbook of Social Choice & Welfare*, volume 2. Elsevier, 2010.

[2] A. Gibbard. Manipulation of Voting Schemes: a General Result. *Econometrica: Journal of the Econometric Society*, pages 587–601, 1973.

[3] M.A. Satterthwaite. Strategy-proofness and Arrow's Conditions: Existence and Correspondence Theorems for Voting Procedures and Social Welfare Functions. *Journal of Economic Theory (JET)*, 10(2):187–217, 1975.

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

[5] Piotr Faliszewski and Ariel D. Procaccia. AI's war on manipulation: Are we winning? *AI MAGAZINE*, 31(4):53–64, 2010.

[6] Nir Ailon. Aggregation of partial rankings, p-ratings and top-m lists. *Algorithmica*, 57(2):284–300, 2010.

[7] Alexander Reffgen. Generalizing the Gibbard–Satterthwaite theorem: partial preferences, the degree of manipulation, and multi-valuedness. *Social Choice and Welfare*, 37(1):39–59, 2011.

[8] Maria Silvia Pini, Francesca Rossi, Kristen Brent Venable, and Toby Walsh. Aggregating Partially Ordered Preferences. *Journal of Logic and Computation*, 19(3):475–502, June 2009.

[9] Edith Elkind and Nisarg Shah. Electing the Most Probable Without Eliminating the Irrational: Voting Over Intransitive Domains. In Nevin L. Zhang and Jin Tian, editors, *UAI*, pages 182–191. AUAI Press, 2014.

[10] Nir Ailon. An Active Learning Algorithm for Ranking from Pairwise Preferences with an Almost Optimal Query Complexity. *Journal of Machine Learning Research*, 13(1):137–164, January 2012.

[11] Xiaoye Jiang, Lek-Heng Lim, Yuan Yao, and Yinyu Ye. Statistical ranking and combinatorial Hodge theory. *Mathematical Programming*, 127(1):203–244, March 2011.

[12] Sahand Negahban, Sewoong Oh, and Devavrat Shah. Iterative ranking from pair-wise comparisons. In *Advances in Neural Information Processing Systems*, pages 2474–2482, 2012.

[13] Xi Chen, Paul N Bennett, Kevyn Collins-Thompson, and Eric Horvitz. Pairwise ranking aggregation in a crowdsourced setting. In *Proceedings of the sixth ACM International Conference on Web Search and Data Mining*, pages 193–202. ACM, 2013.

[14] Arun Rajkumar and Shivani Agarwal. A statistical convergence perspective of algorithms for rank aggregation from pairwise data. In *Proceedings of the 31st International Conference on Machine Learning*, pages 118–126, 2014.

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

[16] Lirong Xia, Michael Zuckerman, Ariel D Procaccia, Vincent Conitzer, and Jeffrey S Rosenschein. Complexity of Unweighted Coalitional Manipulation under Some Common Voting Rules. In *IJCAI*, volume 9, pages 348–352. Citeseer, 2009.

[17] Jessica Davies, Nina Narodytska, and Toby Walsh. Eliminating the Weakest Link: Making Manipulation Intractable? In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, pages 1333–1339. AAAI Press, 2012.

[18] Jessica Davies, George Katsirelos, Nina Narodytska, Toby Walsh, and Lirong Xia. Complexity of and algorithms for the manipulation of Borda, Nanson's and Baldwin's voting rules. *Artificial Intelligence*, 217:20–42, 2014.

[19] Piotr Faliszewski, Edith Hemaspaandra, and Henning Schnoor. Copeland voting: Ties matter. In *Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 983–990. International Foundation for Autonomous Agents and Multiagent Systems, 2008.

[20] Benjamin L Schwartz. Possible winners in partially completed tournaments. *SIAM Review*, 8(3):302–308, 1966.

[21] Walter Kern and Daniël Paulusma. The Computational Complexity of the Elimination Problem in Generalized Sports Competitions. *Discrete Optimization*, 1(2):205–214, November 2004.

[22] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., New York, NY, USA, 1979.

[23] Miroslav Chlebík and Janka Chlebíková. Approximation Hardness for Small Occurrence Instances of NP-hard Problems. In *Algorithms and Complexity*, pages 152–164. Springer, 2003.

[24] Arunava Sen. Another direct proof of the Gibbard–Satterthwaite theorem. *Economics Letters*, 70(3):381–385, 2001.

[25] Navin Aswal, Shurojit Chatterji, and Arunava Sen. Dictatorial domains. *Economic Theory*, 22(1):45–62, 2003.

[26] Shin Sato. Circular domains. *Review of Economic Design*, 14(3-4):331–342, 2010.

[27] Anup Pramanik. Further results on dictatorial domains. *Social Choice and Welfare*, 45(2):379–398, 2015.

[28] John Duggan and Thomas Schwartz. Strategic manipulability without resoluteness or shared beliefs: Gibbard-Satterthwaite generalized. *Social Choice and Welfare*, 17(1):85–93, 2000.

[29] Piotr Faliszewski, Edith Hemaspaandra, and Henning Schnoor. Manipulation of Copeland Elections. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1*, pages 367–374. International Foundation for Autonomous Agents and Multiagent Systems, 2010.

[30] Piotr Faliszewski, Edith Hemaspaandra, Lane A Hemaspaandra, and Jörg Rothe. Llull and Copeland

voting computationally resist bribery and constructive control. *Journal of Artificial Intelligence Research*, pages 275–341, 2009.

[31] Nina Narodytska and Toby Walsh. The Computational Impact of Partial Votes on Strategic Voting. In *Proceedings of the 21st European Conference on Artificial Intelligence (ECAI)*, pages 657–662, 2014.

[32] Zack Fitzsimmons and Edith Hemaspaandra. Complexity of Manipulative Actions When Voting with Ties. In *Algorithmic Decision Theory*, pages 103–119. Springer, 2015.

[33] Vijay Menon and Kate Larson. Complexity of Manipulation in Elections with Top-truncated Ballots. *arXiv preprint arXiv:1505.05900*, 2015.

[34] Vijay Menon and Kate Larson. Reinstating Combinatorial Protections for Manipulation and Bribery in Single-Peaked and Nearly Single-Peaked Electorates. In *AAAI*, 2016.

[35] Kathrin Konczak and Jérôme Lang. Voting procedures with incomplete preferences. In *Proc. IJCAI-05 Multidisciplinary Workshop on Advances in Preference Handling*, volume 20, 2005.

[36] Toby Walsh. Uncertainty in Preference Elicitation and Aggregation. In *AAAI*, volume 7, pages 3–8, 2007.

[37] Nadja Betzler and Britta Dorn. Towards a dichotomy for the possible winner problem in elections based on scoring rules. *Journal of Computer and System Sciences*, 76(8):812–836, 2010.

[38] M. S. Pini, F. Rossi, K. B. Venable, and T. Walsh. Incompleteness and incomparability in preference aggregation: Complexity results. *Artificial Intelligence*, 175(7–8):1272–1289, May 2011.

[39] Dorothea Baumeister and Jörg Rothe. Taking the final step to a full dichotomy of the possible winner problem in pure scoring rules. *Information Processing Letters*, 112(5):186–190, February 2012.

[40] Serge Gaspers, Victor Naroditskiy, Nina Narodytska, and Toby Walsh. Possible and Necessary Winner Problem in Social Polls. In *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-agent Systems*, pages 613–620. International Foundation for Autonomous Agents and Multiagent Systems, 2014.

[41] Lirong Xia and Vincent Conitzer. Determining Possible and Necessary Winners under Common Voting Rules Given Partial Orders. *Journal of Artificial Intelligence Research*, pages 25–67, 2011.

[42] Sergey Brin and Lawrence Page. Reprint of: The anatomy of a large-scale hypertextual web search engine. *Computer networks*, 56(18):3825–3833, 2012.

[43] David C Parkes and Lirong Xia. A complexity-of-strategic-behavior comparison between Schulze's rule and ranked pairs. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI'12)*. American Association for Artificial Intelligence, 2012.