The Price of Neutrality for the Ranked Pairs Method[*]

Markus Brill and Felix Fischer

Abstract

The complexity of the winner determination problem has been studied for almost all common voting rules. A notable exception, possibly caused by some confusion regarding its exact definition, is the method of ranked pairs. The original version of the method, due to Tideman, yields a social preference function that is irresolute and neutral. A variant introduced subsequently uses an exogenously given tie-breaking rule and therefore fails neutrality. The latter variant is the one most commonly studied in the area of computational social choice, and it is easy to see that its winner determination problem is computationally tractable. We show that by contrast, computing the set of winners selected by Tideman's original ranked pairs method is NP-complete, thus revealing a trade-off between tractability and neutrality. In addition, several results concerning the hardness of manipulation and the complexity of computing possible and necessary winners are shown to follow as corollaries from our findings.

1 Introduction

The fundamental problem of social choice theory can be concisely described as follows: given a number of individuals, or *voters*, each having a preference ordering over a set of *alternatives*, how can we aggregate these preferences into a collective, or *social*, preference ordering that is in some sense faithful to the individual preferences? By a preference ordering we here understand a (transitive) ranking of all alternatives, and a function aggregating individual preference orderings into social preference orderings is called a *social preference function* (SPF).[1]

A natural idea to construct an SPF is by letting an alternative a be socially preferred to another alternative b if and only if a majority of voters prefers a to b. However, it was observed as early as the 18th century that this approach might lead to paradoxical situations: the collective preference relation may be cyclic even when all individual preferences are transitive [7].

To remedy this situation, a large number of SPFs have been suggested, together with a variety of criteria that a reasonable SPF should satisfy. *Neutrality* and *anonymity*, for instance, are basic fairness criteria which require, loosely speaking, that all alternatives and all voters are treated equally. Another criterion we will be interested in is the *computational effort* required to evaluate an SPF. Computational tractability of the winner determination problem is obviously a significant property of any SPF: the inability to efficiently compute social preferences would render the method virtually useless, at least for large problem instances that do not exhibit additional structure. As a consequence, computational aspects of preference aggregation have received tremendous interest in recent years (see, e.g., [9, 5, 4]).

In this paper, we study the computational complexity of the *ranked pairs* method [15]. To the best of our knowledge, this question has not been considered before, which is particularly

[*] An earlier version of this paper has appeared in the proceedings of AAAI-2012.

[1] In contrast to a *social welfare function* as studied by Arrow [1], an SPF can output *multiple* social preference orderings with the interpretation that all those rankings are tied for winner. The rationale behind this is to allow for a symmetric outcome when individual preferences are symmetric, like in the case of two individuals with diametrically opposed preferences.

surprising given the extensive literature that is concerned with computational aspects of ranked pairs.[2] A possible reason for this gap might be the confusion of two variants of the method, only one of which satisfies neutrality. In Section 2, we address this confusion and describe both variants. After introducing the necessary notation in Section 3, we show in Section 4 that deciding whether a given alternative is a ranked pairs winner for the neutral variant is NP-complete. Section 5 presents a number of corollaries, and Section 6 discusses variants of the ranked pairs method that are not anonymous.

2 Two Variants of the Ranked Pairs Method

In this section we address the difference between two variants of the ranked pairs method that are commonly studied in the literature. Both variants are anonymous, i.e., treat all voters equally. Non-anonymous variants of the ranked pairs method have been suggested by Tideman [15] and Zavist and Tideman [20], and will be discussed in Section 6.

The ranked pairs method is most easily described as the result of the following procedure. First define a "priority" ordering over the set of all unordered pairs of alternatives by giving priority to pairs with a larger majority margin. Then, construct a ranking of the alternatives by starting with the empty ranking and iteratively considering pairs in order of their priority. When pair $\{a, b\}$ is considered, the ranking is extended by fixing that the majority-preferred alternative precedes the other alternative in the ranking, unless this would create a cycle together with the previously fixed pairs, in which case the opposite precedence between a and b is fixed. Clearly, this procedure is guaranteed to terminate with a complete ranking of all alternatives.

What is missing from the above description is a *tie-breaking rule* for cases where two or more pairwise comparisons have the same support from the voters. This turns out to be a rather intricate issue. In principle, it is possible to employ an arbitrary tie-breaking rule. However, each fixed tie-breaking rule biases the method in favor of some alternative and thereby destroys neutrality.[3] In order to avoid this problem, Tideman [15] originally defined the ranked pairs method to return the set of all those rankings that result from the above procedure for *some* tie-breaking rule.[4] We will henceforth denote this variant by RP.

In a subsequent paper, Zavist and Tideman [20] showed that tie-breaking rules of a certain kind are in fact necessary in order to achieve the property of *independence of clones*, which was the main motivation for introducing the ranked pairs method. While Zavist and Tideman [20] proposed a way to define a tie-breaking rule based on the preferences of a distinguished voter (see Section 6 for details), the variant that is most commonly studied in the literature considers the tie-breaking rule to be exogenously given and fixed for all profiles. This variant of ranked pairs will be denoted by RPT. Whereas RP may output a *set* of rankings, with the interpretation that all the rankings in the set are tied for winner, RPT always outputs a single ranking. In social choice terminology, RP is an *irresolute* SPF, and RPT is a *resolute* one. It is straightforward to see that RP is neutral, i.e., treats all alternatives equally, and that RPT is not. An easy example for the latter statement is the case of two alternatives and two voters who each prefer a different alternative.

Rather than completely ranking all alternatives, it is often sufficient to identify the socially "best" alternatives. This is the purpose of a *social choice function* (SCF). An SCF

[2] Typical problems include the hardness of manipulation [3, 18, 14] and the complexity of computing possible and necessary winners [19, 12].

[3] Neutrality can be maintained if the tie-breaking rule varies with the individual preferences (Section 6).

[4] This definition, sometimes called *parallel universes tie-breaking (PUT)*, can also be used to "neutralize" other voting rules that involve tie-breaking [6]. PUT can be interpreted as a *possible winner* notion: if the ranked pairs method is used with an unknown tie-breaking rule, the PUT version of ranked pairs selects exactly those alternatives that have a chance to be chosen in the actual election.

has the same input as an SPF, but returns alternatives instead of rankings. Each SPF gives rise to a corresponding SCF that returns the top elements of the rankings instead of the rankings themselves, and we will frequently switch between these two settings. Interestingly, deciding whether a given ranking is chosen by an SPF can be considerably easier than deciding whether a given alternative is chosen by the corresponding SCF.

From a computational perspective, RPT is easy: constructing the ranking for a given tie-breaking rule takes time polynomial in the size of the input (see Proposition 1). For RP, however, the picture is different: as the number of tie-breaking rules is exponential, executing the iterative procedure for every single tie-breaking rule is infeasible. Of course, this does not preclude the existence of a clever algorithm that efficiently computes the set of all alternatives that are the top element of some chosen ranking.[5] Our main result states that such an algorithm does not exist unless P equals NP.[6]

3 Preliminaries

For a finite set X, let $\mathcal{L}(X)$ denote the set of all *rankings* of X, where a ranking of X is a complete, transitive, and asymmetric relation on X. The *top element* of a ranking $L \in \mathcal{L}(X)$, denoted by $top(L)$, is the unique element $x \in X$ such that $x \; L \; y$ for all $y \in X \setminus \{x\}$. Furthermore, $\binom{X}{2}$ denotes the set of all two-element subsets of X.

Let $N = \{1, \dots, n\}$ be a set of voters with preferences over a finite set A of alternatives. The preferences of voter $i \in N$ are represented by a ranking $R_i \in \mathcal{L}(A)$. The interpretation of $a \; R_i \; b$ is that voter i strictly prefers a to b. A *preference profile* is an ordered list containing a ranking for each voter.

A *social choice function* (SCF) f associates with every preference profile R a non-empty set $f(R) \subseteq A$ of alternatives. A *social preference function* (SPF) f associates with every preference profile R a non-empty set $f(R) \subseteq \mathcal{L}(A)$ of rankings of A.

An SCF or SPF is *neutral* if permuting the alternatives in the individual rankings also permutes the set of chosen alternatives, or the set of chosen rankings, in the exact same way. Formally, f is neutral if $f(\pi(R)) = \pi(f(R))$ for all preference profiles R and all permutations π of A. An SCF or SPF is *anonymous* if the set of chosen alternatives, or the set of chosen rankings, does not change when the voters are permuted.

For a given preference profile $R = (R_1, \dots, R_n)$ and two distinct alternatives $a, b \in A$, the *majority margin* $m_R(a, b)$ is defined as the difference between the number of voters who prefer a to b and the number of voters who prefer b to a, i.e.,

$$m_R(a, b) = |\{i \in N : a \; R_i \; b\}| - |\{i \in N : b \; R_i \; a\}|.$$

Thus, $m_R(b, a) = -m_R(a, b)$ for all distinct $a, b \in A$.

The resolute variant of the ranked pairs method takes as input a preference profile R and a *tie-breaking rule* $\tau \in \mathcal{L}(A \times A)$. It constructs a priority ordering of $\binom{A}{2}$ by ordering all two-element subsets by the size of their majority margin, using τ to break ties: $\{a, b\}$ has priority over $\{c, d\}$ if $|m_R(a, b)| > |m_R(c, d)|$, or if $|m_R(a, b)| = |m_R(c, d)|$ and $(a, b) \; \tau \; (c, d)$.[7] The priority ordering is then used to obtain a ranking $\succ_\tau^R \in \mathcal{L}(A)$ by way of the following iterative procedure. Initialise $\succ_\tau^R$ as the empty relation. Iteratively consider the pair $\{a, b\}$

[5] As the number of chosen *rankings* might be exponential, it immediately follows that computing all of them requires exponential time in the worst case.

[6] A similar discrepancy can be observed for an SCF known as the *Banks set* [2]. Whereas Woeginger [17] has proven that computing Banks winners is NP-complete, Hudry [10] has shown that an arbitrary Banks winner can be found efficiently.

[7] Here we assume without loss of generality that the pairs (a, b) and (c, d) are ordered in such a way that $(a, b) \; \tau \; (b, a)$ and $(c, d) \; \tau \; (d, c)$.

with the highest priority among all pairs in $\binom{A}{2}$ that have not been considered so far. There are two cases.

- *Case 1:* $|m_R(a,b)| \neq 0$. Without loss of generality assume $m_R(a,b) > 0$. If the relation $\succ_\tau^R \cup \{(a,b)\}$ is acyclic, the (ordered) pair (a,b) is added to the relation $\succ_\tau^R$. Otherwise, the pair (b,a) is added to $\succ_\tau^R$.
- *Case 2:* $|m_R(a,b)| = 0$. Without loss of generality assume $(a,b)\ \tau\ (b,a)$. If the relation $\succ_\tau^R \cup \{(a,b)\}$ is acyclic, the pair (a,b) is added to the relation $\succ_\tau^R$. Otherwise, the pair (b,a) is added to $\succ_\tau^R$.

After all pairs in $\binom{A}{2}$ have been considered, $\succ_\tau^R$ is a ranking of A. The resolute variant of ranked pairs, interpreted as an SCF, returns the top element of $\succ_\tau^R$.

Definition 1. $\mathit{RPT}(R,\tau) = \{\mathit{top}(\succ_\tau^R)\}$.

RPT depends on the choice of τ, and it is not neutral. Tideman [15] defined an irresolute and neutral variant that chooses all alternatives that are at the top of $\succ_\tau^R$ for *some* tie-breaking rule τ.

Definition 2. $\mathit{RP}(R) = \{a \in A :$ *there exists* $\tau \in \mathcal{L}(A \times A)$ *such that* $a = \mathit{top}(\succ_\tau^R)\}$.

The alternatives in $\mathsf{RP}(R)$ are called *ranked pairs winners for R.* In the SPF setting, RP returns the rankings $\{\succ_\tau^R : \tau \in \mathcal{L}(A \times A)\}$, which are henceforth called *ranked pairs rankings for R.*

We will work with an alternative characterization of ranked pairs rankings that was introduced by Zavist and Tideman [20]. Given a preference profile R, a ranking L of A, and two alternatives a and b, we say that *a attains b through L* if there exists a sequence of distinct alternatives $a_1, a_2, \dots, a_t$, where $t \geq 2$, such that $a_1 = a$, $a_t = b$, $a_i\ L\ a_{i+1}$, and

$$m_R(a_i, a_{i+1}) \geq m_R(b,a) \text{ for all } i \text{ with } 1 \leq i < t.$$

In this case, we will say that a attains b *via* $(a_1, a_2, \dots, a_t)$. A ranking L is called a *stack* if for any pair of alternatives a and b it holds that $a\ L\ b$ implies that a attains b through L.

Lemma 1 (Zavist and Tideman [20])**.** *A ranking of A is a ranked pairs ranking if and only if it is a stack.*

It follows that an alternative is a ranked pairs winner if and only if it is the top element of a stack.

4 Complexity of Winner Determination

We are now ready to study the computational complexity of RP. We first consider the SPF setting and observe that finding and checking ranked pairs rankings is easy. This also provides an efficient way to find *some* ranked pairs winner, i.e., some alternative that is chosen in the SCF setting. The problem of deciding whether a *particular* alternative is a ranked pairs winner, on the other hand, turns out to be NP-complete. Finally, we extend the hardness result to a variant of the winner determination problem that asks for unique winners.

4.1 Ranked Pairs Rankings

It can easily be seen that an arbitrary ranked pairs ranking can be found efficiently.

Proposition 1. *Finding a ranked pairs ranking is in P.*

Proof. We fix some arbitrary tie-breaking rule $\tau \in \mathcal{L}(A \times A)$ and compute $\succ_\tau^R$, which, by definition, is a ranked pairs ranking. When constructing $\succ_\tau^R$, in each round we have to check whether the addition of a pair (a, b) to the relation $\succ_\tau^R$ creates a cycle. This can efficiently be done with a depth-first search. □

Deciding whether a given ranking is a ranked pairs ranking is also feasible in polynomial time, by checking whether the given ranking is a stack.

Proposition 2. *Deciding whether a given ranking is a ranked pairs ranking is in P.*

Proof. By Lemma 1, it suffices to check whether the given ranking L is a stack. This reduces to checking, for every pair (a, b) with $a\ L\ b$, whether a attains b through L. Let a and b with $a\ L\ b$ be given, and define $w = m_R(b, a)$. We construct a directed graph with vertex set A as follows. For all $x, y \in A$, there is an edge from x to y if and only if $x\ L\ y$ and $m_R(x, y) \geq w$. It is easily verified that a attains b through L if and only if there exists a path from a to b in this graph. The latter property can be efficiently checked with a depth-first search. Since the number of pairs in L is polynomial, this proves the statement. □

4.2 Ranked Pairs Winners

We now consider the SCF setting. As every ranked pairs ranking yields a ranked pairs winner, Proposition 1 immediately implies that an arbitrary element of $\mathsf{RP}(R)$ can be found efficiently.

Corollary 1. *Finding a ranked pairs winner is in P.*

Deciding whether a given alternative is a ranked pairs winner, on the other hand, turns out to be NP-complete.

Theorem 1. *Deciding whether a given alternative is a ranked pairs winner is NP-complete.*

Membership in NP follows from Proposition 2. For hardness, we give a reduction from the NP-complete Boolean satisfiability problem (SAT, see, e.g., [13]). An instance of SAT consists of a Boolean formula $\varphi = C_1 \wedge \cdots \wedge C_k$ in conjunctive normal form over a finite set $V = \{v_1, \ldots, v_m\}$ of variables. Denote by $X = \{v_1, \overline{v}_1, \ldots, v_m, \overline{v}_m\}$ the set of all *literals*, where a literal is either a variable or its negation. Each *clause* C_j is a set of literals. An *assignment* $\alpha \subseteq X$ is a subset of the literals with the interpretation that all literals in α are set to "true." Assignment α is *valid* if $\ell \in \alpha$ implies $\overline{\ell} \notin \alpha$ for all $\ell \in X$, and α *satisfies* clause C_j if $C_j \cap \alpha \neq \emptyset$. A valid assignment that satisfies all clauses of φ is a *satisfying assignment* for φ, and a formula that has a satisfying assignment is called *satisfiable.*

For a particular Boolean formula $\varphi = C_1 \wedge \cdots \wedge C_k$ over a set $V = \{v_1, \ldots, v_m\}$ of variables, we will construct a preference profile R_φ over a set A_φ of alternatives such that a particular alternative $d \in A_\varphi$ is a ranked pairs winner for R_φ if and only if φ is satisfiable.

Let us first define the set A_φ of alternatives. For each variable $v_i \in V$, $1 \leq i \leq m$, there are four alternatives v_i, $\overline{v}_i$, v_i', and $\overline{v}_i'$. For each clause C_j, $1 \leq j \leq k$, there is one alternative y_j. Finally, there is one alternative d for which we want to decide membership in $\mathsf{RP}(R_\varphi)$.

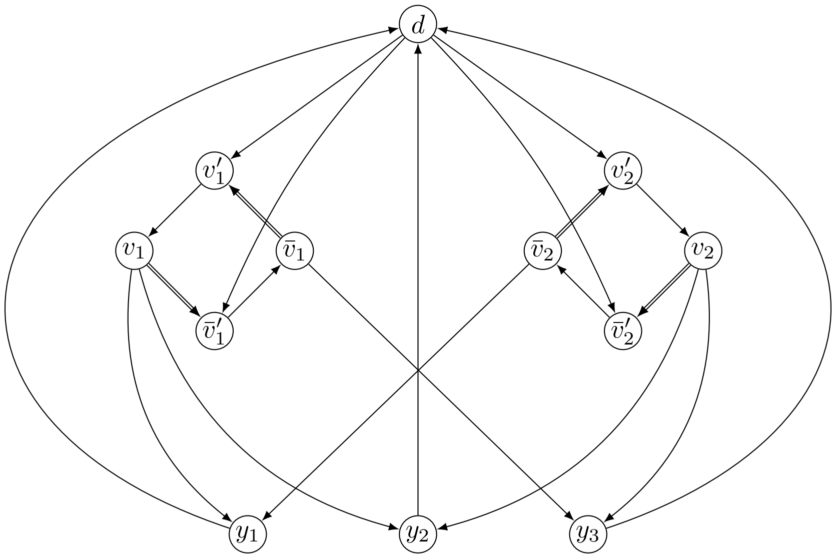

Figure 1: Graphical representation of $m_{R_\varphi}(\cdot,\cdot)$ for the Boolean formula $\varphi = \{v_1, \bar{v}_2\} \wedge \{v_1, v_2\} \wedge \{\bar{v}_1, v_2\}$. The relation $\succ^2$ is represented by arrows, and $\succ^4$ is represented by double-shafted arrows. For all pairs (a, b) that are not connected by an arrow, we have $m(a, b) = m(b, a) = 0$.

Instead of constructing R_φ explicitly, we will specify a number $m(a, b)$ for each pair $(a, b) \in A_\varphi \times A_\varphi$. Debord [8] has shown that there exists a preference profile R such that $m_R(a, b) = m(a, b)$ for all a, b, as long as $m(a, b) = -m(b, a)$ for all a, b and all the numbers $m(a, b)$ have the same parity.[8] In order to conveniently define $m(\cdot,\cdot)$, the following notation will be useful: for a natural number w, $a \succ^w b$ denotes setting $m(a, b) = w$ and $m(b, a) = -w$.

For each variable $v_i \in V$, $1 \le i \le m$, let $v_i \succ^4 \bar{v}_i' \succ^2 \bar{v}_i \succ^4 v_i' \succ^2 v_i$. For each clause C_j, $1 \le j \le k$, let $v_i \succ^2 y_j$ if variable $v_i \in V$ appears in clause C_j as a positive literal, and $\bar{v}_i \succ^2 y_j$ if variable v_i appears in clause C_j as a negative literal. Finally let $y_j \succ^2 d$ for $1 \le j \le k$ and $d \succ^2 v_i'$ and $d \succ^2 \bar{v}_i'$ for $1 \le i \le m$. For all pairs (a, b) for which $m(a, b)$ has not been specified so far, let $m(a, b) = m(b, a) = 0$. An example is shown in Figure 1.

As $m(a, b) \in \{-4, -2, 0, 2, 4\}$ for all $a, b \in A_\varphi$, Debord's theorem guarantees the existence of a preference profile R_φ with $m_{R_\varphi}(a, b) = m(a, b)$ for all $a, b \in A_\varphi$, and such a profile can in fact be constructed efficiently, i.e., in polynomial time.

The following two lemmata show that alternative d is a ranked pairs winner for R_φ if and only if the formula φ is satisfiable.

Lemma 2. *If $d \in \mathit{RP}(R_\varphi)$, then φ is satisfiable.*

Proof. Assume that d is a ranked pairs winner for R_φ and let L be a stack with $top(L) = d$. Consider an arbitrary j with $1 \le j \le k$. As L is a stack and $d\ L\ y_j$, d attains y_j through L, i.e., there exists a sequence $P_j = (a_1, a_2, \ldots, a_t)$ with $a_1 = d$ and $a_t = y_j$ such that $a_i\ L\ a_{i+1}$ and $m(a_i, a_{i+1}) \ge 2$ for all i with $1 \le i < t$. If d attains y_j via several sequences, fix one of them arbitrarily.

[8] Also see the article by Le Breton [11].

The definition of $m(\cdot,\cdot)$ implies that

$$\begin{aligned} P_j &= (d,\overline{\ell}',\overline{\ell},\ell',\ell,y_j) \quad \text{or} \\ P_j &= (d,\ell',\ell,y_j), \end{aligned}$$

where ℓ is some literal. The former is in fact not possible because $m(\ell,\overline{\ell}') = 4$ implies that $\overline{\ell}'$ does *not* attain ℓ through L. Therefore, each P_j is of the form $P_j = (d,\ell',\ell,y_j)$ for some $\ell \in X$.

Now define assignment α as the set of all literals that are contained in one of the sequences $P_j, 1 \le j \le k$, i.e., $\alpha = X \cap (\bigcup_{j=1}^{k} P_j)$. We claim that α is a satisfying assignment for φ.

In order to show that α is valid, suppose there exists a literal $\ell \in X$ such that both ℓ and $\overline{\ell}$ are contained in α. This implies that there exist i and j such that d attains y_i via $P_i = (d,\ell',\ell,y_i)$ and d attains y_j via $P_j = (d,\overline{\ell}',\overline{\ell},y_j)$. In particular, $\ell' \; L \; \ell$ and $\overline{\ell}' \; L \; \overline{\ell}$. It follows that either $\ell' \; L \; \overline{\ell}$ or $\overline{\ell}' \; L \; \ell$, as otherwise $(\ell,\overline{\ell}',\overline{\ell},\ell')$ would form an L-cycle, contradicting the transitivity of L. However, neither does ℓ' attain $\overline{\ell}$ through L, nor does $\overline{\ell}'$ attain ℓ through L, a contradiction.

In order to see that α satisfies φ, consider an arbitrary clause C_j. As d attains y_j via $P_j = (d,\ell',\ell,y_j)$ and $m(y_j,d) = 2$, we have that $m(\ell,y_j) \ge 2$. By definition of $m(\cdot,\cdot)$, this implies that $\ell \in C_j$. □

Lemma 3. *If φ is satisfiable, then $d \in \mathsf{RP}(R_\varphi)$.*

Proof. Assume that φ is satisfiable and let α be a satisfying assignment. Let $V_i = \{v_i,\overline{v}_i,v_i',\overline{v}_i'\}$, $1 \le i \le m$, and $Y = \{y_1,y_2,\dots,y_k\}$. We define a ranking L of A_φ as follows, using $B \; L \; C$ as shorthand for $b \; L \; c$ for all $b \in B$ and $c \in C$.

- For all $1 \le i \le m$, let $d \; L \; V_i$ and $V_i \; L \; Y$.
- For all $1 \le i < j \le m$, let $V_i \; L \; V_j$.
- For the definition of L within V_i, we distinguish two cases. If $v_i \in \alpha$, i.e., if v_i is set to "true" under α, let $\overline{v}_i \; L \; v_i' \; L \; v_i \; L \; \overline{v}_i'$. If, on the other hand, $v_i \notin \alpha$, let $v_i \; L \; \overline{v}_i' \; L \; \overline{v}_i \; L \; v_i'$.
- Within Y, define L arbitrarily.

We now prove that L is a stack. For each pair (a,b) with $a \; L \; b$, we need to verify that a attains b through L. If $m(b,a) \le 0$, it is easily seen that a attains b through L. We can therefore assume that $m(b,a) > 0$. By definition of L and $m(\cdot,\cdot)$, a particular such pair (a,b) satisfies either

$$\begin{aligned} &a = d \text{ and } b \in Y, \text{ or} \\ &a,b \in V_i \text{ for some } i \text{ with } 1 \le i \le m. \end{aligned}$$

First consider a pair of the former type, i.e., $(a,b) = (d,y_j)$ for some j with $1 \le j \le k$. As α satisfies C_j, there exists $\ell \in C_j$ with $\ell \in \alpha$. Consider the sequence $P_j = (d,\ell',\ell,y_j)$. As $m(y_j,d) = 2$ and $d \succ^2 \ell' \succ^2 \ell \succ^2 y_j$, d attains y_j via P_j.

Now consider a pair of the latter type, i.e., $a,b \in V_i$ for some i with $1 \le i \le m$. Assume that $v_i \in \alpha$ and, therefore, $\overline{v}_i \; L \; v_i' \; L \; v_i \; L \; \overline{v}_i'$. The only non-trivial case is the pair $(\overline{v}_i,\overline{v}_i')$ with $\overline{v}_i \; L \; \overline{v}_i'$ and $m(\overline{v}_i',\overline{v}_i) = 2$. But $\overline{v}_i$ attains $\overline{v}_i'$ via $(\overline{v}_i,v_i',v_i,\overline{v}_i')$ because $\overline{v}_i \succ^4 v_i' \succ^2 v_i \succ^4 \overline{v}_i'$. The case $v_i \notin \alpha$ is analogous.

We have shown that L is a stack. Lemma 1 now implies that $d \in \mathsf{RP}(R_\varphi)$, which completes the proof. □

Combining Lemma 2 and Lemma 3, and observing that both A_φ and R_φ can be constructed efficiently, completes the proof of Theorem 1.

4.3 Unique Winners

An interesting variant of the winner determination problem concerns the question whether a given alternative is the *unique* winner for a given preference profile. Despite its similarity to the original winner determination problem, this problem is sometimes considerably easier.[9] For RP, the picture is different: verifying unique winners is not feasible in polynomial time, unless P equals coNP.

Theorem 2. *Deciding whether a given alternative is the unique ranked pairs winner is coNP-complete.*

Proof. Membership in coNP follows from the observation that for every "no" instance there is a stack whose top element is different from the alternative in question.

For hardness, we modify the construction from Section 4.2 to obtain a reduction from the problem UNSAT, which asks whether a given Boolean formula is *not* satisfiable. For a Boolean formula φ, define $A'_\varphi = A_\varphi \cup \{d^*\}$, where d^* is a new alternative and A_φ is defined as in Section 4.2. R'_φ is defined such that $d \succ^2 d^*$ and $d^* \succ^4 a$ for all $a \in A_\varphi \setminus \{d\}$. Within A_φ, R'_φ coincides with R_φ. We show that $\mathsf{RP}(R'_\varphi) = \{d^*\}$ if and only if φ is unsatisfiable.

For the direction from left to right, assume for contradiction that $\mathsf{RP}(R'_\varphi) = \{d^*\}$ and φ is satisfiable. Consider a satisfying assignment α and let L be the ranking of A_φ defined in the proof of Lemma 3. Define the ranking L' of A'_φ by

$$L' = L \cup \{(d, d^*)\} \cup \{(d^*, a) : a \in A_\varphi \setminus \{d\}\}.$$

That is, L' extends L by inserting the new alternative d^* in the second position. As in the proof of Lemma 3, it can be shown that L' is a stack. It follows that $top(L') = d \in \mathsf{RP}(R'_\varphi)$, contradicting the assumption that $\mathsf{RP}(R'_\varphi) = \{d^*\}$.

For the direction from right to left, assume for contradiction that φ is unsatisfiable and $\mathsf{RP}(R'_\varphi) \neq \{d^*\}$. Then there exists a tie-breaking rule τ such that $top(\succ_\tau^{R'_\varphi}) = a \neq d^*$. From the definition of R'_φ it follows that $a = d$, as $d^* \succ^4 b$ for all $b \in A_\varphi \setminus \{d\}$ and there are no $\succ^4$-cycles. By the same argument as in the proof of Lemma 2, it can be shown that φ is satisfiable, contradicting our assumption. □

5 New Proofs for Old and New Results

In this section we briefly consider computational problems other than winner determination. We show that our findings imply several hardness results, some of which are already known. We also point out some errors in the literature that are due to the assumption that winner determination for ranked pairs is in P. By Theorem 1, this assumption is incorrect unless P=NP. All results concern the neutral variant RP, and we refer to the respective papers for formal definitions of the computational problems.

An alternative a is a *possible winner* for a partially specified preference profile R if there exists a completion R' of R such that a is a winner for R'. It is a *necessary winner* if it is a winner for every completion of R. Both the possible and the necessary winner problem have a variant that requires an alternative to be the *unique* winner for the completions.

Corollary 2. *Computing possible ranked pairs winners is NP-complete. Computing possible* unique *ranked pairs winners is both NP-hard and coNP-hard.*

[9] The Banks set, discussed in Footnote 6, constitutes an example: although deciding membership is NP-complete in general, it can be checked in polynomial time whether an alternative is the *unique* Banks winner. The reason for the latter is that an alternative is the unique Banks winner if and only if it is a Condorcet winner.

Proof. NP-completeness of the non-unique variant was already shown by Xia and Conitzer [18]. Membership in NP holds because for every "yes" instance there exists a completion and a tie-breaking rule that yields the alternative in question. Hardness also follows from Theorem 1, because the possible winner problem is equivalent to the winner determination problem in the special case when the preference profile is completely specified.

NP-hardness of the unique variant was shown by Xia and Conitzer [18]; coNP-hardness follows from Theorem 2, because the possible unique winner problem is equivalent to the unique winner determination problem in the special case when the preference profile is completely specified. Xia and Conitzer [18] in fact claimed NP-completeness, but their argument for membership in NP assumes that winner determination is in P. □

Corollary 3. *Computing necessary ranked pairs winners is both NP-hard and coNP-hard. Computing necessary* unique *ranked pairs winners is coNP-complete.*

Proof. Hardness of the non-unique variant for coNP was shown by Xia and Conitzer [18]; NP-hardness follows from Theorem 1, because the necessary winner problem is equivalent to the winner determination problem in the special case when the preference profile is completely specified. Xia and Conitzer [18] in fact claim coNP-completeness, but their argument for membership in coNP assumes that winner determination is in P.

Completeness of the unique variant for coNP was shown by Xia and Conitzer [18]. Membership in coNP holds because for every "no" instance there is a completion and a tie-breaking rule that produces a different winner. Hardness also follows from Theorem 2, because the necessary unique winner problem is equivalent to the unique winner determination problem in the special case when the preference profile is completely specified. □

The *unweighted coalitional manipulation* (UCM) problem asks whether it is possible for a group of voters to cast their votes in such a way that a distinguished alternative becomes a (non-unique or unique) winner.

Corollary 4. *The non-unique UCM problem under ranked pairs is NP-complete. The unique UCM problem under ranked pairs is both NP-hard and coNP-hard.*

Proof. NP-completeness of the non-unique variant was already shown by Xia et al. [19].[10] Membership in NP holds because for every "yes" instance there is a preference profile for the manipulators and a tie-breaking rule that outputs the alternative in question. Hardness also follows from Theorem 1, because the non-unique UCM problem with zero manipulators is equivalent to the winner determination problem.

NP-hardness of the unique variant was shown by Xia et al. [19]; coNP-hardness follows from Theorem 2, because the unique UCM problem with zero manipulators is equivalent to the unique winner determination problem. Xia et al. [19] in fact claimed NP-completeness, but their argument for membership in NP assumes that winner determination is in P. □

6 Non-Anonymous Variants

As mentioned in Section 2, Tideman [15] and Zavist and Tideman [20] suggested ways to use the preferences of a distinguished voter, say, a chairperson, to render the ranked pairs method resolute. There are essentially two ways to achieve this, which differ in the point in time when ties are broken. For the sake of simplicity, we only consider the SCF setting in this section.

[10] The proof of Theorem 4.1 by Xia et al. [19] actually works for both RP and RPT (Xia, personal communication, March 29, 2012).

The *a priori* variant uses the preferences of the chairperson to construct a tie-breaking rule $\tau \in \mathcal{L}(A \times A)$, which is then used to compute $\mathsf{RPT}(\cdot, \tau)$. The *a posteriori* variant first computes $\mathsf{RP}(\cdot)$ and then chooses the alternative from this set that is most preferred by the chairperson. Both variants are neutral: if the alternatives are permuted in each ranking, including the ranking of the chairperson, the tie-breaking rule and thus the chosen alternative will change accordingly.

Whereas the *a priori* variant is a special case of RPT and therefore efficiently computable, the *a posteriori* variant is intractable by the results in Section 4. It follows that neutrality and tractability can be reconciled at the expense of anonymity. By moving to *non-deterministic* SCFs, one can even regain anonymity: choosing the chairperson for the *a priori* variant uniformly at random results in a procedure that is neutral, anonymous, and tractable, for appropriate generalizations of anonymity and neutrality to the case of non-deterministic SCFs. The winner determination problem for the *a posteriori* variant remains intractable when the chairperson is chosen randomly.

7 Conclusion

We have studied the complexity of the ranked pairs method. While *some* ranked pairs winner is easy to find, deciding whether a given alternative is a winner turns out to be NP-complete. If one is interested in ranked pairs rankings, both problems are computationally easy.

From a practical point of view, the ranked pairs method is easier than most other intractable SCFs. The reason is that the *expected* number of ties among two or more pairs is rather small. This is particularly true when the number of voters is large compared to the number of alternatives, which is the case in many realistic settings. It is therefore to be expected that ranked pairs winners are easy to compute *on average* for most reasonable distributions of individual preferences.

Our results reveal a trade-off between neutrality and tractability in the context of the ranked pairs method: while the efficiently computable variant RPT fails neutrality, the neutral variant RP is intractable. A very similar trade-off can be observed for the *single transferable vote* rule [6, 16].

We have finally discussed variants of the ranked pairs method that achieve neutrality at the expense of anonymity, by using individual preferences to break ties. The tractability of those variants depends on the point in time ties are broken.

Acknowledgments This material is based on work supported by the Deutsche Forschungsgemeinschaft under grants BR 2312/10-1 and FI 1664/1-1. We thank anonymous referees at AAAI and COMSOC for valuable comments and suggestions.

References

[1] K. J. Arrow. *Social Choice and Individual Values.* New Haven: Cowles Foundation, 1st edition, 1951. 2nd edition 1963.

[2] J. S. Banks. Sophisticated voting outcomes and agenda control. *Social Choice and Welfare*, 3:295–306, 1985.

[3] N. Betzler, S. Hemmann, and R. Niedermeier. A multivariate complexity analysis of determining possible winners given incomplete votes. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, pages 53–58, 2009.

[4] F. Brandt, V. Conitzer, and U. Endriss. Computational social choice. In G. Weiß, editor, *Multiagent Systems.* MIT Press, 2nd edition, 2012. Forthcoming.

[5] V. Conitzer. Making decisions based on the preferences of multiple agents. *Communications of the ACM*, 53(3):84–94, 2010.

[6] V. Conitzer, M. Rognlie, and L. Xia. Preference functions that score rankings and maximum likelihood estimation. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, pages 109–115, 2009.

[7] Marquis de Condorcet. *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix.* Imprimerie Royale, 1785. Facsimile published in 1972 by Chelsea Publishing Company, New York.

[8] B. Debord. Caractérisation des matrices des préférences nettes et méthodes d'agrégation associées. *Mathématiques et sciences humaines*, 97:5–17, 1987.

[9] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, and J. Rothe. A richer understanding of the complexity of election systems. In S. Ravi and S. Shukla, editors, *Fundamental Problems in Computing: Essays in Honor of Professor Daniel J. Rosenkrantz.* Springer-Verlag, 2009.

[10] O. Hudry. A note on "Banks winners in tournaments are difficult to recognize" by G. J. Woeginger. *Social Choice and Welfare*, 23:113–114, 2004.

[11] M. Le Breton. On the uniqueness of equilibrium in symmetric two-player zero-sum games with integer payoffs. *Économie publique*, 17(2):187–195, 2005.

[12] S. Obraztsova and E. Elkind. On the complexity of voting manipulation under randomized tie-breaking. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 319–324. AAAI Press, 2011.

[13] C. H. Papadimitriou. *Computational Complexity.* Addison-Wesley, 1994.

[14] D. Parkes and L. 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).* AAAI Press, 2012. Forthcoming.

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

[16] B. A. Wichmann. Tie breaking in STV. *Voting Matters*, 19:1–5, 2004.

[17] G. J. Woeginger. Banks winners in tournaments are difficult to recognize. *Social Choice and Welfare*, 20:523–528, 2003.

[18] L. Xia and V. Conitzer. Determining possible and necessary winners under common voting rules given partial orders. *Journal of Artificial Intelligence Research*, 41:25–67, 2011.

[19] L. Xia, M. Zuckerman, A. D. Procaccia, V. Conitzer, and J. S. Rosenschein. Complexity of unweighted coalitional manipulation under some common voting rules. In *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, pages 348–353. AAAI Press, 2009.

[20] T. M. Zavist and T. N. Tideman. Complete independence of clones in the ranked pairs rule. *Social Choice and Welfare*, 6(2):167–173, 1989.

Markus Brill
Institut für Informatik
Technische Universität München
85748 Garching bei München, Germany
Email: brill@in.tum.de

Felix Fischer
Statistical Laboratory
University of Cambridge
Cambridge CB3 0WB, UK
Email: fischerf@statslab.cam.ac.uk