Incompleteness and Incomparability in Preference Aggregation: Complexity Results

Maria Silvia Pini[1], Francesca Rossi[1], K. Brent Venable[1] and Toby Walsh[2]

[1] University of Padova, Italy
{mpini,frossi,kvenable}@math.unipd.it
[2] NICTA and UNSW, Sydney, Australia.
tw@cse.unsw.edu.au

Abstract. We consider how to combine the preferences of multiple agents despite the presence of incompleteness and incomparability in their preference orderings. An agent's preference ordering may be incomplete because, for example, there is an ongoing preference elicitation process. It may also contain incomparability, which can be useful, for example, in multi-criteria scenarios. We focus on the problem of computing the *possible* and *necessary* winners, that is, those outcomes which can be or always are the most preferred for the agents. Possible and necessary winners are useful in many scenarios, including preference elicitation. First we show that computing the sets of possible and necessary winners is in general a difficult problem as it is providing a good approximation of such sets. Then we identify sufficient conditions, related to general properties of the preference aggregation function, where such sets can be computed in polynomial time. Finally, we show how possible and necessary winners can be used to focus preference elicitation.

1 Introduction

We consider a multi-agent setting where each agent specifies their preferences by means of an ordering over the possible outcomes. A pair of outcomes can be ordered, incomparable, in a tie, or the relationship between them may not yet be specified. Incomparability and incompleteness represent very different concepts. Outcomes may be incomparable because the agent does not wish very dissimilar outcomes to be compared. For example, we might not want to compare a biography with a novel as the criteria along which we judge them are just too different. Outcomes can also be incomparable because the agent has multiple criteria to optimize. For example, we might not wish to compare a faster but more expensive laptop with a slower and cheaper one. Incompleteness, on the other hand, represents simply an absence of knowledge about the relationship between certain pairs of outcomes. Incompleteness arises naturally when we have not fully elicited an agent's preferences or when agents have privacy concerns which prevent them revealing their complete preference ordering.

As we wish to aggregate together the agents' preferences into a single preference ordering, we must modify preference aggregation functions to deal with incompleteness. One possibility is to consider all possible ways in which the incomplete preference orders can be consistently completed. In each possible completion, preference aggregation may give different optimal elements (or *winners*). This leads to the idea of the *possible winners* (those outcomes which are winners in at least one possible completion) and the *necessary winners* (those outcomes which are winners in all possible completions) [9].

While voting theory has been mainly interested in possibility or impossibility results about social choice or social welfare functions, recently there has been some interest also in computational properties of preference aggregation [12, 10, 9, 7]. It has also been noted that the complexity of deciding whether there is a manipulation in an election is closely related to the complexity of computing possible winners [9, 6].

In this paper we start by considering the complexity of computing the necessary and the possible winners. We show that both tasks are hard in general, even to approximate.

Then we identify sufficient conditions that assure tractability. Such conditions concern properties of the preference aggregation function, such as monotonicity and independence to irrelevant alternatives (IIA) [1], which are natural properties to require.

Possible and necessary winners are useful in many scenarios including preference elicitation [4]. For example, elicitation is over when the set of possible winners coincides with that of the necessary winners [7]. However, recognizing when such a condition is satisfied is hard in general. In the last part of the paper we show that, if the preference aggregation function is IIA, preference elicitation can focus just on the incompleteness concerning those outcomes which are possible and necessary winners, allowing us to ignore all other outcomes and to complete preference elicitation in polynomial time.

In this paper we extend the results presented in [11], by giving complexity results concerning the computation of the exact and approximate sets of possible and necessary winners and by giving sufficient conditions on the preference aggregation function, that allow to compute in polynomial time not only the exact set of the necessary winners, but also the exact set of the possible winners.

2 Basic notions

Preferences. We assume that each agent's preferences are specified via a (possibly incomplete) partial order with ties (IPO) over the set of possible outcomes, that we will denote by Ω. An incomplete partial order is a partial order where some relation between pairs of outcomes is unknown. Given two outcomes A and B, an agent will specify exactly one of the following: $A < B$, $A > B$, $A = B$, $A \sim B$, or $A?B$, where $A \sim B$ means that A and B are incomparable, and $A?B$ that the relation between A and B is unknown, this means that it can be any element of $\{=, >, <, \sim\}$.

Example 1. Given outcomes A, B, and C, an agent may state preferences such as $A > B$, $B \sim C$, and $A > C$, or also just $A > B$ and $B \sim C$. However, an agent cannot state preferences such as $A > B$, $B > C$, $C > A$, or also $A > B$, $B > C$, $A \sim C$ since neither are POs.

Profiles. A *profile* is a sequence of n partial orders $p_1, \ldots, p_n$ over outcomes, one for each agent $i \in \{1, \ldots, n\}$, describing the preferences of the agents. An *incomplete profile* is a sequence in which one or more of the partial orders is incomplete.

Social welfare and preference aggregation. Social welfare functions [1] are functions from profiles to partial orders with ties. Given a social welfare function f, we define a corresponding *preference aggregation function*, written pa_f, which is a function from incomplete profiles to sets of partial orders with ties (POs). Precisely, given an incomplete profile $ip = (ip_1, \ldots, ip_n)$, where the ip_i's are IPOs, consider all the profiles, say $p_1, \ldots, p_k$, obtained from ip by replacing any occurrence of ? in the ip_i's with either $<, >$, $=$, or $\sim$ which is consistent with a partial order. Let us then set $pa_f(ip) = \{f(p_1), \ldots, f(p_k)\}$. This set will be called the *set of results* of f on profile ip.

Example 2. Consider the Pareto social welfare function f defined as follows [1]: given a profile p, for any two outcomes A and B, if all agents say $A > B$ or $A = B$ and at least one says $A > B$ in p, then $A > B \in f(p)$; if all agents say $A = B$ in p, then $A = B \in f(p)$; otherwise, $A \sim B \in f(p)$. In Figure 1 we show an example with three agents and three outcomes A, B, and C.

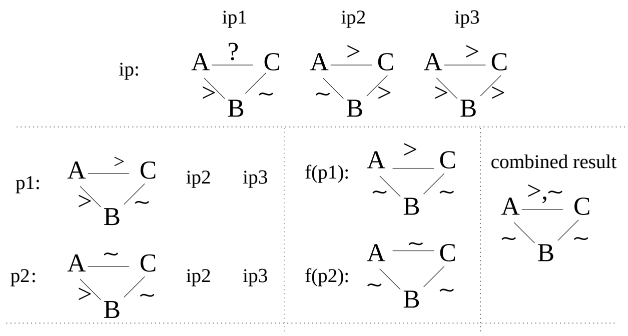

Fig. 1. An incomplete profile ip, its completions p_1 and p_2, the results $f(p_1)$ and $f(p_2)$, and the combined result $cr(f, ip)$.

Necessary and possible winners. We extend to the case of partial orders the notions of possible and necessary winners presented in [9] in the case of total orders. Given a social welfare function f and an incomplete profile ip, we define *necessary winners* of f given ip as all those outcomes which are maximal elements in all POs in $pa_f(ip)$. A necessary winner must be a winner, no matter how incompleteness is resolved in the incomplete profile. Analogously, the *possible winners* are all those outcomes which are maximal elements in at least one of the POs in $pa_f(ip)$. A possible winner is a winner in at least one possible completion of the incomplete profile.

We will write $NW(f, ip)$ and $PW(f, ip)$ for the set of necessary and possible winners of f on profile ip. We will sometimes omit f and/or ip, and just write NW and PW when they will be obvious or irrelevant.

Example 3. In Example 2, A and B are the necessary winners, since they are top elements in all POs $f(p_i)$, for all $i = 1, 2$. C is a possible winner since it wins in $f(p_2)$.

Combined result. Unfortunately, the set of results can be exponentially large. We will therefore also consider a compact representation that is polynomial in size. This may throw away information by compacting together results into a single combined result. Given a social welfare function f and an incomplete profile ip, consider a graph, whose nodes are the outcomes, and whose arcs are labeled by non-empty subsets of $\{<, >, =, \sim\}$. Label l is on the arc between outcomes A and B if there exists a PO in $pa_f(ip)$ where A and B are related by l. This graph will be called the *combined result* of f on ip, and will be denoted by $cr(f, ip)$. If an arc is labeled by set $\{<, >, =, \sim\}$, we will say that it is *fully incomplete*. Otherwise, we say that it is *partially incomplete*. The set of labels on the arc between A and B will be called $rel(A, B)$.

Example 4. The combined result for Example 2 is shown in Figure 1.

3 Possible and Necessary Winners

In this section we show that computing the set of necessary and possible winners of a social welfare function is, in general, NP-hard even if we restrict ourselves to incomplete but *total* orders. We will consider the following, well known, voting rule.

Single Transferable Vote. In the STV rule each voter provides a total order on candidates and, initially, an individual's vote is allocated to their most preferred candidate. The *quota* of the election is the minimum number of votes necessary to get elected. If only one candidate is to be elected then the quota is $\lfloor n/2 \rfloor + 1$, where n is the number of voters. If no candidate exceeds the quota, then, the candidate with the fewest votes is eliminated, and his votes are equally distributed among the second choices of the voters who had selected him as first choice. In what follows we consider STV elections in which some total orders, provided by the voters, are incomplete.

In general, given an incomplete profile and a candidate a, we say PossibleWinner holds iff a is a possible winner of the election.

Theorem 1. `PossibleWinner` *is NP-complete.*

Proof. In fact, membership of NP follows by giving a completion of the profile in which a wins. Completeness follows from the result that EFFECTIVE PREFERENCE (determining if a particular candidate can win an election with one vote unknown) for STV is NP-complete [2]

This result allows us to conclude that, in general, finding possible winners of an election is difficult. However, it should be noticed that for many rules used in practice including some positional scoring rules [9], answering `PossibleWinner` is polynomial. The complexity of computing possible winners is related to the complexity of deciding if there is a manipulation in an election [9]. For instance, it is NP-complete to determine for the Borda, Copeland, Maximin and STV rules if a coalition can cast weighted votes to ensure a given winner [6]. It follows therefore that with weighted votes, `PossibleWinner` is NP-hard for these rules.

Given an incomplete profile and a candidate a, we say NECESSARYWINNER holds iff a is a necessary winner of the election.

Theorem 2. `NecessaryWinner` *is coNP-complete.*

Proof. The complement problem is in NP since we can show membership by giving a completion of the profile in which some b different to a wins. To show completeness, we give a reduction from EFFECTIVE PREFERENCE with STV in which a appears at least once in first place in one vote. This restricted form of EFFECTIVE PREFERENCE is NP-complete [2]. Consider an incomplete profile Π in which n votes have been cast, a has at least one first place vote, one vote remains unknown, and we wish to decide if a can win. We construct a new election from Π with n new additional votes, and one new candidate b. We put b at the top of each of these new votes, and rank the other candidates in any order within these n votes. We place b in last place in the original n votes, except for one vote where a is in first place (by assumption, one such vote must exist) where we place b in second place and shift all other candidates down. We observe that b will survive till the last round as b has at least $n-1$ votes and no other candidate can have as many till the last round. We also observe that if a remains in the election, then the score given to each candidate by STV remains the same as in the original election so the candidates are eliminated in the same order up till the point a is eliminated. If a is eliminated before the last round, the second choice vote for b is transferred. Since b now has $n+1$ votes, b is unbeatable and must win the election. If a survives, on the other hand, to the last round, we can assume b is ranked at the bottom of the unknown vote. All the other candidates but a and b have been eliminated so a has n votes and is unbeatable. Hence, if a is not a possible winner in the original election, b is the necessary winner of this new election. Thus determining the necessary winner of this new election decides if a is a possible winner of the original election.

Given these results, we might wonder if it is easy to compute a reasonable approximation of the sets of possible and necessary winners. Unfortunately this is not the case. The reduction described in the proof of previous theorem shows that we cannot approximate the set of possible winners within a factor of two. In fact, we can show that we cannot approximate efficiently the set of possible winners within any constant factor.

Theorem 3. *It is NP-hard to return a superset of the possible winners, PW^* in which we guarantee $|PW^*| < k|PW|$ for some given positive integer k.*

Proof. We again give a reduction from EFFECTIVE PREFERENCE for STV in which a appears at least once in first place in one vote. Consider an incomplete profile Π in which n votes have been cast, a has at least one first place vote, one vote remains unknown, and we wish to decide if a can win. We construct a new election from Π. We make k copies of Π. In the ith copy Π_i, we subscript each candidate with the integer i. We add n new additional votes, and one new candidate b. We put b at the top of each of these new votes, and rank all the other candidates except a_i in any order within these n votes. The ranking of the candidates a_i is left unknown but beneath b. In each Π_i, we place b in last place except for one vote where a_i is in first place (by assumption, one such vote must exist) where we place b in second place and shift all other candidates down. Finally, for each candidate in Π_j not in Π_i except for a_j, we rank then in any order at the bottom of the votes in Π_i. The ranking of the candidates a_i is again left unknown but beneath b. We observe that b will survive till all but one candidate has been eliminated from one of the Π_i. We also observe that if a_i remains in the election, then the score given to each candidate by STV remains the same as in the original election so the candidates in Π_i are eliminated in the same order up till the point a_i is eliminated. Suppose a cannot win the original election. Then a_i will always be eliminated before the final round. The second choice vote for b is transferred. Since b now has at least $n + 1$ votes, b is unbeatable and must win the election. Suppose, on the other hand, that a can win the original election. Then a_i can survive to be the last remaining candidate in Π_i. We can assume b is ranked at the bottom of the unknown votes of all the candidates with an index i and above all the candidates with an index j different to i. Thus a_i has n votes. If we have the corresponding ranking in the other unknown votes, a_j for $j \neq i$ will also survive. As b has only $n - 1$ votes, b will be eliminated. It is now possible for any of the candidates, a_i where $1 \leq i \leq k$ to win depending on how exactly the a_i are ranked in the different votes. Thus the set of possible winners is $\{a_i \mid 1 \leq i \leq k\}$ plus b if a is not a necessary winner in the original election. Hence, if a is a possible winner in the original election, the size of the set of possible winners is greater than or equal to k, whilst if it is not, the set is of size 1. If we know that $|PW^*| < k|PW|$, then $|PW^*| < k$ guarantees that $|PW| = 1$, b is the necessary winner and hence that a is not a possible winner in the original election.

Similarly, we cannot approximate efficiently the set of necessary winners within some fixed ratio.

Theorem 4. *It is NP-hard to return a subset of the necessary winners, NW^* in which we guarantee $|NW^*| > \frac{1}{k}|NW|$ whenever $|NW| > 0$ for some given positive integer k.*

Proof. In the reduction used in the last proof, $|NW| = 1$ if a is a possible winner in the original election and 0 otherwise. But if $|NW| = 1$ and $|NW^*| > \frac{1}{k}|NW|$

then $|NW^*| = 1$. Hence $|NW^*| = 1$ iff a is a possible winner. Thus, the size of NW^* will determine if a is possible winner.

4 Combined result

We now consider the problem of computing the combined result. We show that, while in general it is difficult, there are some restrictions which allow us to compute an approximation of the combined result in polynomial time. In the next section, we will show how it is possible to compute the set of possible and necessary winners starting from this approximation to the combined result.

Theorem 5. *Given an incomplete profile, determining if a label is in the combined result for STV is NP-complete.*

Proof. In fact, a polynomial witness is a completion of the incomplete profile. To show completeness, we use a polynomial number of calls to this problem to determine if a given candidate is a possible winner.

From this result we immediately get the following corollary.

Corollary 1. *Given an incomplete profile and a social welfare function, computing the combined result is NP-hard.*

We now introduce some properties of preference aggregation functions which allow us to compute an upper approximation to the combined result in polynomial time. We recall that the set of labels of an arc between A and B in the combined result is called $rel(A, B)$.

The first property we consider is independence to irrelevant alternatives (IIA). A social welfare function is said to be IIA when, for any pair of outcomes A and B, the ordering between A and B in the result depends only on the relation between A and B given by the agents [1]. Many preference aggregation functions are IIA, and this is a desirable property which is related to the notion of fairness in voting theory [1]. Given a function which is IIA, to compute the set $rel(A, B)$, we just need to ask each agent their preference over the pair A and B, and then use f to compute all possible results between A and B. However, if agents have incompleteness between A and B, f has to consider all the possible completions, which is exponential in the number of such agents.

Assume now that f is also monotonic. We say that an outcome B *improves with respect to* another outcome A if the relationship between A and B does not move left along the following sequence: $>, \geq, (\sim \text{ or } =), \leq, <$. For example, B improves with respect to A if we pass from $A \geq B$ to $A \sim B$. A social welfare function f is *monotonic* if for any two profiles p and p' and any two outcomes A and B passing from p to p' B improves with respect to A in one agent i and $p_j = p'_j$ for all $j \neq i$, then passing from $f(p)$ to $f(p')$ B improves with respect to A.

Consider now any two outcomes A and B. To compute $rel(A, B)$ under IIA and monotonicity, again, since f is IIA, we just need to consider the agents' preferences over the pair A and B. However, now we don't need to consider all possible completions for all agents with incompleteness between A and B, but just two completions: $A < B$ and $A > B$. Function f will return a result for each of these two completions, say AxB and AyB, where $x, y \in \{<, >, =, \sim\}$. Since f is monotonic, the results of all the other completions will necessarily be between x and y in the ordering $>$, $\geq$, ($\sim$ or $=$), $\leq$, $<$. By taking all such relations, we obtain a superset of $rel(A, B)$, that we call $rel^*(A, B)$. In fact, monotonicity of f assures that, if we consider profile $A < B$ and we get a certain result, then considering profiles where A is in a better position w.r.t. B (that is, $A > B$, $A = B$, or $A \sim B$), will give an equal or better situation for A in the result. Thus we have obtained an approximation of the combined result, that we call $cr^*(f, ip)$. We will now give a characterization of this approximation of the combined result.

Theorem 6. *Given two outcomes A and B, $rel^*(A, B) \supseteq rel(A, B)$. Moreover, if $rel^*(A, B) = \{<, >, \sim, =\}$, then either $rel^*(A, B) = rel(A, B)$ or $rel^*(A, B) - rel(A, B) = \{\sim, =\}$.*

By following the procedure informally described above, this approximation can be computed polynomially, since we only need to consider two completions.

Theorem 7. *Given a preference aggregation function f which is IIA and monotonic, and an incomplete profile ip, computing $cr^*(f, ip)$ is polynomial in the number of agents.*

5 Computing possible and necessary winners

We will now show how to determine the possible and necessary winners, given $cr^*(f, ip)$. Consider the arc between an outcome A and an outcome C in $cr^*(f, ip)$. Then, if this arc has the label $A < C$, then A is not a necessary winner, since there is an outcome C which is better than A in some result. If this arc *only* has the label $A < C$, then A is not a possible winner since we must have $A < C$ in all results. Moreover, consider all the arcs between A and every other outcome C. Then, if no such arc has label including $A < C$, then A is a necessary winner. Notice, however, that in general, even if none of the arcs connecting A have just a single label $A < C$, A could not be possible winner. A could be better than some outcomes in every completion, but there might be no completion where it is better than all of them. We will show that this is not the case if f is IIA and monotonic.

We now define Algorithm 1, which, given $cr^*(f, ip)$, computes NW and PW, in polynomial time.

Theorem 8. *Given $cr^*(f, ip)$, Algorithm 1 terminates in $O(m^2)$ time, where $m = |\Omega|$, returning $N = NW$ and $P = PW$.*

Proof. Algorithm 1 considers, in the worst case, each arc exactly once, thus we have $O(m^2)$.

Algorithm 1: Computing NW and PW

Input: $cr^*(f, ip)$, where f: IIA and monotonic preference aggregation function and ip: incomplete profile;
Output: P, N: sets of outcomes;
$P \leftarrow \Omega$;
$N \leftarrow \Omega$;
foreach $A \in \Omega$ **do**
  if $\exists\, C \in \Omega$ *such that* $\{<\} \subseteq rel^*(A, C)$ **then**
    $N \leftarrow N - \{A\}$;
  if $\exists\, C \in \Omega$ *such that* $\{<\} = rel^*(A, C)$ **then**
    $P \leftarrow P - \{A\}$;
return P, N;

N=NW. By construction of $cr^*(f, ip)$, $<\notin rel^*(A, C)$ iff $<\notin rel(A, C)$. By Algorithm 1, $A \in N$ iff $\forall C, <\notin rel\{A, C\}$, and this implies that there is no result in which there exists an outcome C that beats A. Thus, $A \in NW$. On the contrary, $A \in NW$ iff $A \not< C, \forall C \in \Omega$, for all results, from which, $A \in N$.

$P = PW$. By Algorithm 1, an outcome is in P iff there is no other outcome which beats it in all results. Thus, $PW \subseteq P$. To show the other inclusion we consider $A \in P$ and we construct a completion of ip such that A wins in its result. First, let us point out that for any outcome A, $A \in P$ iff $\not\exists C \in \Omega$, $rel(A, C) = \{<\}$. If $\forall C \in \Omega$, $<\notin rel(A, C)$, then we already know that A is NW, and thus a PW. Assume now that $\exists C \in \Omega$ such that $\{<\} \subset rel(A, C)$. Consider now any arc from A to another outcome C', labeled with more than one relation. If $\forall C' \in \Omega - \{C\}$, $|rel^*(A, C')| = 1$, then all the arcs from A, except AC, are labeled with exactly one label from the set: $\{>, \sim, =\}$. In such a case, we can safely set AC to any of its labels other than $<$, since there is, for sure, a result with that labeling. Moreover, in such a result A is a winner. Assume, instead, that there is at least an outcome C' such that $|rel^*(A, C')| > 1$. This means that there is at least an agent which has not declared his preference on AC' and that such preference cannot be induced by transitivity closure. We replace $A?C'$ with $A > C'$ everywhere in the profile, we perform the transitive closure of all the modified $IPOs$, and we apply f. We will prove that such a procedure will never force to choose label $<$ on AC. After the procedure, $rel(A, C')$ will contain exactly one label from the set: $\{>, \sim, =\}$. Let us now consider $rel(C', C)$. We consider only the cases in which $|rel(C', C)| = 1$, since they are the most restrictive and they imply all others in terms of transitivity. Let us assume that, after the procedure, $A = C'$. If $rel^*(C', C) = \{<\}$, then, $C' < C$ in all results. Due to monotonicity, we know that, $rel^*(A, C') = \{<, =\}$ or $rel^*(A, C') = \{=\}$. By transitivity, this would force $rel(A, C) = \{<\}$. However, this is not possible since $A \in P$. This allows us to conclude that $(rel^*(C', C) \cap \{>, \sim, =\}) \neq \emptyset$ and any of such additional labels together with $A = C'$ can never force $A < C$. Clearly, if $A > C'$ or $A \sim C'$, there is no labeling of $C'C$ which can force $A < C$. It should be noticed that any available choice on $C'C$ can always be made safely due to the fact that the function is IIA and that the transitive closure of the profiles has already ruled out inconsistent choices. By iterating the procedure until every ? in the incomplete profile is replaced, we can construct a result of the function in which A is a winner.

An example of a preference aggregation function which is both IIA and monotonic is the Pareto rule, described in Example 2. Another example is the Lex rule, in which agents are ordered and, given any two outcomes A and B, the relation between A and B in the result is the relation given by the first agent in the order that does not declare a tie between A and B. A third example is the approval voting rule, for which the tractability result has been already proven in [9] since it is a positional scoring rule.

6 Preference elicitation

One use of necessary and possible winners is in eliciting preferences [4]. Preference elicitation is the process of asking queries to agents in order to determine their preferences over outcomes.

At each stage in eliciting agents' preferences, there is a set of possible and necessary winners. When $NW = PW$, preference elicitation can be stopped since we have enough information to declare the winners, no matter how the remaining incompleteness is resolved [7]. At the beginning, NW is empty and PW contains all outcomes. As preferences are declared, NW grows and PW shrinks. At each step, an outcome in PW can either pass to NW or become a loser.

Determining the winners. In those steps where PW is still larger than NW, we can use these two sets to guide preference elicitation and avoid useless work. In fact, to determine if an outcome $A \in PW - NW$ is a loser or a necessary winner, it is enough to ask agents to declare their preferences over all pairs involving A and another outcome, say B, in PW. In fact, any outcome outside PW is a loser, and thus is dominated by at least one possible winner.

If the preference aggregation function is IIA, then all those pairs (A, B) with a defined preference for all agents can be avoided, since they will not help in determining the status of outcome A. Moreover, IIA allows us to consider just one profile when computing the relations between A and B in the result, and assures that the result is a precise relation, that is, either $<$, or $>$, or $=$, or $\sim$. In the worst case, we need to consider all such pairs. To determine all the winners, we thus need to know the relations between A and B for all $A \in PW - NW$ and $B \in PW$. Again, there are examples where all such pairs must be considered.

We can thus use Algorithm 2, which in $O(|PW|^2)$ steps eliminates enough incompleteness to determine the winners. At each step, the algorithm asks each agent to express its preferences on a pair of outcomes (via procedure $ask(A, B)$) and aggregates such preferences via function f. If function f is polynomially computable, the whole computation is polynomial in the number of agents and outcomes.

Theorem 9. *If f is IIA and polynomially computable, then determining the set of winners via preference elicitation is polynomial in the number of agents and outcomes.*

Using the results of the previous sections, under certain conditions we know how to compute efficiently the necessary winners and the possible winners. Thus Algorithm 2 can be given in input the outputs of Algorithm 1.

Algorithm 2: Winner determination

```
Input: PW, NW: sets of outcomes; f: preference aggregation function;
Output: W: set of outcomes;
wins: bool;
P ← PW; N ← NW;
while P ≠ N do
    choose A ∈ P − N;
    wins ← true; P_A ← P − {A};
    repeat
        choose B ∈ P_A;
        if ∃ an agent such that A?B then
            ask(A,B);
            compute f(A,B);
        if f(A,B) = (A > B) then
            P ← P − {B};
        if f(A,B) = (A < B) then
            P ← P − {A}; wins ← false;
        P_A ← P_A − {B};
    until f(A,B) ≠ (A < B) or P_A ≠ ∅ ;
    if wins = true then
        N ← N ∪ {A};
W ← N;
return W;
```

It should be noticed that deciding when elicitation is over, that is checking if $P = N$, is hard in general since, in [7] such a result has been proven for STV.

7 Related and future work

In [9] preference aggregation functions for combining incomplete total orders are considered. Compared to our work, we permit both incompleteness and incomparability, while they allow only for incompleteness. Second, they consider social choice functions which return the (non-empty) set of winners. Instead, we consider social welfare functions which return a complete partial order. Social welfare functions give a finer grained view of the result. Third, they consider specific voting rules like the Borda procedure whilst we have focused on general properties that ensure tractability.

The results presented in this paper can be useful not just for combining preferences from multiple agents, but also for combining multiple conflicting preferences from a single agent. Recent work addressing the combination of multiple complex preferences is presented in [5] and [8].

We plan to consider the addition of constraints to agents' preferences. This means that preference aggregation must take into account the feasibility of the outcomes. Thus possible and necessary winners must now be feasible.

It is also important to consider compact knowledge representation formalisms to express agents' preferences, such as CP-nets [12] and soft constraints [3]. Possible and necessary winners should then be defined directly from such compact representations, and preference elicitation should concern statements allowed in the representation language.

Finally, a possibility distribution over the completions of an incomplete preference relation between two outcomes can be used to provide additional information when computing possible and necessary winners.

References

1. K. J. Arrow, A. K. Sen, and K. Suzumara. *Handbook of Social Choice and Welfare.* North-Holland, Elsevier, 2002.
2. J.J. Bartholdi and J.B. Orlin. Single transferable vote resists strategic voting. *Social Choice and Welfare*, 8(4):341–354, 1991.
3. S. Bistarelli, U. Montanari, and F. Rossi. Semiring-based constraint solving and optimization. *Journal of the ACM*, 44(2):201–236, mar 1997.
4. L. Chen and P. Pu. Survey of preference elicitation methods. Technical Report IC/200467, Swiss Federal Institute of Technology in Lausanne (EPFL), 2004.
5. Jan Chomicki. Semantic optimization of preference queries. In *CDB*, pages 133–148, 2004.
6. V. Conitzer and T. Sandholm. Complexity of manipulating elections with few candidates. In *AAAI/IAAI*, pages 314–319, 2002.
7. V. Conitzer and T. Sandholm. Vote elicitation: Complexity and strategy-proofness. In *Proc. AAAI/IAAI 2002*, pages 392–397, 2002.
8. W. Kießling. Preference queries with sv-semantics. In *11th International Conference on Management of Data (COMMAD 2005)*, pages 15–26. Computer Society of India, 2005.
9. K. Konczak and J. Lang. Voting procedures with incomplete preferences. In *Proc. IJCAI-05 Multidisciplinary Workshop on Advances in Preference Handling*, 2005.
10. J. Lang. Logical preference representation and combinatorial vote. *Annals of Mathematics and Artificial Intelligence*, 42(1):37–71, 2004.
11. M. S. Pini, F. Rossi, K. Brent Venable, and T. Walsh. Computing possible and necessary winners from incomplete partially-ordered preferences. In *Poster paper in ECAI-06*, 2006.
12. F. Rossi, K. B. Venable, and T. Walsh. mCP nets: representing and reasoning with preferences of multiple agents. In *AAAI-2004*, pages 322–327, 2004.