Possible and necessary winners in voting trees: majority graphs vs. profiles

Maria Silvia Pini
University of Padova
Padova, Italy
mpini@math.unipd.it

Francesca Rossi
University of Padova
Padova, Italy
frossi@math.unipd.it

Kristen Brent Venable
Universita' di Padova
Padova, Italy
kvenable@math.unipd.it

Toby Walsh
NICTA and UNSW
Sydney, Australia
toby.walsh@nicta.com.au

ABSTRACT

Given the preferences of several agents over a common set of candidates, voting trees can be used to select a candidate (the winner) by a sequence of pairwise competitions modelled by a binary tree (the agenda). The majority graph compactly represents the preferences of the agents and provides enough information to compute the winner. When some preferences are missing, there are various notions of winners, such as the possible winners (that is, winners in at least one completion) or the necessary winners (that is, winners in all completions). In this generalized scenario, we show that using the majority graph to compute winners is not correct, since it may declare as winners candidates that are not so. Nonetheless, the majority graph can be used to compute efficiently an upper or lower approximation of the correct set of winners.

Categories and Subject Descriptors

I.2.11 [**Computing methodologies**]: Artificial Intelligence—*Distributed Artificial Intelligence*

General Terms

Algorithms, Theory

Keywords

Preferences, incompleteness, necessary winners, voting trees

1. INTRODUCTION

Voting is a simple and natural mechanism to aggregate the preferences of multiple agents. Results like those of Gibbard and Sattertwhaite demonstrate that, under weak assumptions like an election of more than two candidates, no voting rule is ideal. Many different voting rules, with different properties, have therefore been proposed. Voting trees are a general method that can implement many such rules [2]. A voting tree is a binary tree (called the agenda) where the leaves are labelled with the candidates in the election, the internal nodes are labelled with the winner of the pairwise comparison between the children, and the root wins the overall election. Voting trees can implement, for example, any voting rule over three candidates [3]. Voting trees have therefore been studied as a general, abstract model for decision-making among multiple agents (see, for example, [10, 9, 11]).

In practice, we often want to make decisions despite the presence of uncertainty. Uncertainty can come in different forms. There may be uncertainty about the votes [4, 12, 1, 13]. We may, for example, have only partially elicited preferences, or we may only have partial knowledge about the votes of other agents. There may also be uncertainty about the voting rule itself. For instance, with voting trees, there may be uncertainty about the agenda [7]. This may be because the agenda is not yet fixed (for example, the agenda is to be chosen at a later date by means of a random draw as in the World Cup), or the agenda is not announced in advance (to impede manipulation), or the chair may still be in a position to change the agenda (in order to manipulate the result). We are therefore interested in computing the results of an election with a voting tree where there is uncertainty in the vote and/or in the voting tree.

When some preferences are missing, there are various notions of winners, such as the possible winners (that is, winners in at least one completion) or the necessary winners (that is, winners in all completions) [4]. When computing possible and necessary winners, we can work from the incomplete votes or the majority graph. The majority graph summarizes the votes in the form of a directed graph where there is a directed arc between two candidates iff a majority of the agents prefer the first candidate over the second. We can also consider whether the agenda is fixed or unknown. Previously, Lang et al. [5] studied the problem of computing possible and necessary winners from the majority graph. In [8], Pini et al. compared these results to the problem of computing possible and necessary winners from incomplete profiles. However, this work left open several important relationships between the different types of winners, which we close here.

To be precise, we consider all notions of winners in the setting of incomplete preferences or uncertain agenda (possible Condorcet, necessary Condorcet, possible Schwartz, necessary Schwartz, possible, and necessary winners) and study

Cite as: Possible and necessary winners in voting trees: majority graphs vs. profiles, Maria Silvia Pini, Francesca Rossi, Kristen Brent Venable, and Toby Walsh, *Proc. of 10th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, Tumer, Yolum, Sonenberg and Stone (eds.), May, 2–6, 2011, Taipei, Taiwan, pp. XXX-XXX.

whether each of these sets of winners is computable by just looking at the majority graph, without having to consider the whole profile. In other words, we study if the set of winners computed from the majority graph coincides with that computed from the profile. The reason to do this is that computing winners from the majority graph takes polynomial time, even when there are exponential many completions.

For some notions of winners, the literature already tells us if the two sets coincide or not [8]. In fact, we know that, when considering simple voting trees (that is, agendas where each candidate appears exactly once as a leaf), possible Condorcet winners and necessary Condorcet winners can be safely computed from the majority graph, while this is not so for possible Schwartz winners. We first close all open questions about simple voting trees, showing that unfortunately equality does not hold in general for all remaining notions of winners. We then examine the more general setting of voting trees, where we show that the results for simple voting trees are maintained. This means that, both in simple voting trees and in general voting trees, reasoning with the majority graph always produces correct results only when we look for possible and necessary Condorcet winners. When we aim to find other kinds of winners, working with the majority graph gives an upper or lower approximation of the correct set of winners. The situation is even worse for voting trees, since some polynomial algorithms that correctly return winners from the majority graph in the simple voting tree setting do not work well for voting trees, since they may return incorrect responses also w.r.t. the majority graph-based notion of winners. However, what they return is still a lower or upper approximation of the correct set of winners.

2. BASIC NOTIONS

We now give the basic notions for (simple) voting trees [5, 8, 6].

2.1 Preferences, profiles, and majority graphs

We assume that each agent's *preferences* are specified by a strict total order (TO), that is, by an asymmetric, transitive and complete order, on a set of m candidates. The candidates are taken from a set Ω, and they represent the possible options over which agents vote.

A *profile* P on Ω is a collection of n strict total orders over Ω, i.e., $P = (P_1, \ldots, P_n)$, where P_i is the preference relation of agent i. An *incomplete preference relation* $>$ on Ω is a strict order on Ω, that is, a transitive and irreflexive relation on Ω. An *incomplete profile* on Ω is a collection $P = (P_1, \ldots, P_n)$ of incomplete preference relations on Ω. Let $P = (P_1, \ldots, P_n)$ be an incomplete profile over a set of candidates Ω, a completion R of P is a tuple $(R_1, \ldots, R_n)$ such that every R_i is a strict total order on Ω containing P_i. For simplicity, we assume that the number of the agents is odd.

Given an (incomplete) profile P, the *majority graph* $M(P)$ induced by P is the directed graph whose set of vertices is Ω, and where an edge from A to B (denoted by $A >_m B$) denotes a strict majority of voters who prefer A to B. A majority graph is said to be complete if, for any two vertices, there is a directed edge between them, and fully incomplete if there are no edges. Also, if $M(P)$ is incomplete, the set of all complete majority graphs extending $M(P)$ corresponds to a (possibly proper) superset of the set of complete majority graphs induced by all possible completions of P.

2.2 Voting trees

Given a set of candidates, the *simple voting tree rule* (resp., *voting tree rule*) is defined by a binary tree with one candidate per leaf. Each candidate appears exactly once in the leaves (resp., each candidate may appear more than once in the leaves). Each internal node represents the candidate that wins the pairwise election between the node's children. The winner of every pairwise election is computed by the majority rule, where A beats B iff there is a majority of votes stating $A > B$. The candidate at the root of the tree is the overall winner. Given a complete profile, candidates which win for every (simple) voting tree are called *Condorcet winners* and candidates which win for at least one (simple) voting tree are called *Schwartz winners*.

2.3 Notions of winners

Various kinds of winners have been defined from incomplete profiles and from incomplete majority graphs [5, 8, 6].

DEFINITION 1. *Let P be an incomplete profile and A a candidate.*

- *A is a* possible Schwartz winner *for P (i.e., $A \in PossS(P)$) iff there exists a completion of P and a simple voting tree for which A wins;*
- *A is the* necessary Schwartz winner *for P (i.e., $A \in NecS(P)$) iff for every completion of P there is a simple voting tree for which A wins;*
- *A is a* possible Condorcet winner *for P (i.e., $A \in PossC(P)$) iff there is a completion of P such that A is a winner for every simple voting tree;*
- *A is the* necessary Condorcet winner *for P (i.e., $A \in NecC(P)$) iff for every completion of P, and for every simple voting tree, A is a winner.*

When the voting tree is given, the following two notions of winners can also be considered [6, 12].

DEFINITION 2. *Let P be an incomplete profile, A a candidate, and T a simple voting tree.*

- *A is a* possible winner *for P and T (i.e., $A \in Poss(P,T)$) iff there exists a completion of P for which A wins in T;*
- *A is the* necessary winner *for P and T (i.e., $A \in Nec(P,T)$)) iff, for every completion of P, A wins in T.*

When the profile is complete, necessary and possible Condorcet winners coincide. The same holds also for necessary and possible Schwartz winners, and for necessary and possible winners.

The definitions of winners given above can be defined also from incomplete majority graphs [5, 8]. The only difference is such definitions consider the completions of the incomplete majority graph and not those of the incomplete profile.

2.4 Majority graph vs. profile

When the profile is complete, the possible and necessary versions of the same notion of winner collapse, and reasoning from the majority graph is enough for correctly computing the winner. However, when the profile is incomplete, considering the majority graph rather than the profile may give different results. A majority graph may have completions that do not correspond to any completion of the profile. Thus, for certain notions of winners, the possible/necessary winners for the incomplete profile do not coincide with the possible/necessary winner for the incomplete majority graph.

Of course, the correct notion is that defined from the profile, but in some cases the two notions of winners coincide, so we can safely consider only the majority graph. This is more convenient since the majority graph is a compact representation of the profile. Reasoning over this structure is therefore more efficient.

In Table 1 we summarize what is known about the relationship between the set of winners from an incomplete profile P and the set of winners from the incomplete majority graph $M(P)$, both in simple voting trees and in voting trees.

	SVT	**VT**
Possible Schwartz winners	$\neq$ [8]	?
Necessary Schwartz winners	?	?
Possible Condorcet	= [8]	?
Necessary Condorcet	= [8]	?
Possible winners	?	?
Necessary winners	?	?

Table 1: State of the art about winners computed from majority graph or profile, for simple voting trees (SVT) and voting trees (VT).

In words, we know that possible Condorcet winners and necessary Condorcet winners can be correctly computed from the majority graph, while this is not so for possible Schwartz winners. However, we don't know anything about the other notions of winners, except the obvious subset inclusion that comes from the simple observation that the majority graph may have more completions than the profile. More precisely, every notion in its "possible" version and related to the profile denotes a subset of the same notion related to the majority graph. For example, $Poss(P,T) \subseteq Poss(M(P),T)$. Vice versa, every notion in its "necessary" version and related to the majority graph denotes a subset of the same notion related to the profile. For example, $Nec(M(P),T) \subseteq Nec(P,T)$.

2.5 Computing majority graph winners

All the sets of winners for the majority graph can be computed in polynomial time for simple voting trees [5, 6].

Given a simple voting tree T and an incomplete majority graph G, algorithm *Win*, presented in [6], computes, the set of possible winners for G and T. This algorithm (see the pseudocode below) recursively takes in input a simple voting tree T, an incomplete majority graph G, and it returns a set of candidates W, which is the set of possible winners for G and T. If $root(T)$ is not empty, and both $left(T)$ (i.e., the left subtree of T) and $right(T)$ (i.e., the right subtree of T) are empty, then the algorithm returns $label(root(T))$ (i.e., the candidate which labels the root of T). Otherwise, the set of winners at the root of T is the set of all candidates who are possible winners in the left (resp., right) branch of T and who beat at least one candidate who is a possible winner in the right (resp., left) branch of T.

Algorithm *StrongWin* [6] runs algorithm Win on T and G, and just checks if the output is a single candidate. If so, it declares it the necessary winner, otherwise it returns the empty set as there is then no necessary winner.

Algorithm 1: *Win*

```
Input: T: a simple voting tree, G: an incomplete majority graph;
Output: W: set of candidates;
if T contains only one node then
    W ← label(root(T))
else
    W_1 ← Win(left(T), G);
    W_2 ← Win(right(T), G);
    W ← ∅;
    foreach (s, t) ∈ W_1 × W_2 do
        if s >_m t then
            W ← W ∪ {s}
        else
            if t >_m s then
                W ← W ∪ {t}
            else
                W ← W ∪ {s, t}
return W
```

3. WINNERS IN SIMPLE VOTING TREES

We now show that the set of the necessary Schwartz winners (resp., possible winners, necessary winners) for an incomplete profile P may be different from the set of the necessary Schwartz winners (resp., possible winners, necessary winners) for the incomplete majority graph $M(P)$ in simple voting trees. These results close all the open questions in the simple voting tree column of Table 1.

3.1 Necessary Schwartz winners

As noted above, a necessary Schwartz winner from an incomplete majority graph $M(P)$ is always a necessary Schwartz winner from the incomplete profile P [8]. More precisely, let P be an incomplete profile. Then $NecS(M(P)) \subseteq NecS(P)$. However, in general, the opposite does not hold.

THEOREM 1. *There is an incomplete profile P such that $NecS(P) \neq NecS(M(P))$.*

PROOF. To show the result, we give an incomplete profile P and a candidate A such that $A \in NecS(P)$ and $A \notin NecS(M(P))$.

Let $\Omega = \{A, A_1, B_1, B_2, B_3\}$. Assume that we have 5 agents and that the incomplete profile P is defined as follows:

- agent 1: $(A_1 > B_2 > B_3,\ A > B_1)$;
- agent 2: $(B_2 > B_3 > A_1 > B_1 > A)$;
- agent 3: $(A > A_1 > B_3 > B_1 > B_2)$;
- agent 4: $(B_1 > A > B_2 > B_3 > A_1)$;
- agent 5: $(B_3 > B_1 > B_2 > A > A_1)$.

Given this profile, the corresponding majority graph has the following edges:

- $A >_m A_1$, $B_1 >_m A$, $B_1 >_m B_2$,
- $B_2 >_m B_3$, $B_2 >_m A_1$, $B_3 >_m B_1$, $B_3 >_m A_1$.

By Theorem 8 of [5], which states that $A \in NecS(M(P))$ iff there is a path from A to every other candidate in $M(P)$, since there is no path from A to B_1 in $M(P)$, then $A \notin NecS(M(P))$. However, $A \in NecS(P)$. In fact, for every completion of P, there is a tree where A wins. Notice that in P only the first agent expresses incomplete preferences. Therefore, the completions of P are as many as the completions of the first agent's preferences. Such completions can be partitioned in two types: those where the first agent puts A_1 above A and those where the first agent puts A above A_1:

- In every completion of P where A_1 is above A in the first agent's preferences, we have, by transitivity, $A_1 > B_1$ for this agent. This yields a majority of agents stating $A_1 > B_1$. Therefore, for every completion of this kind, the corresponding majority graph has the edge $A_1 >_m B_1$. It is possible to see that A wins in the tree where first B_2 plays against B_3, the winner (that is, B_2) plays against B_1, the winner (that is, B_1) plays against A_1, and finally the winner (that is, A_1) plays against A.
- In every completion of P where A is above A_1 in the first agent's preferences, we have, by transitivity, $A > B_2$ for this agent. This yields a majority of agents stating $A > B_2$. Therefore, for every completion of this kind, the corresponding majority graph has the edge $A >_m B_2$. It is possible to see that A wins in the tree where first B_3 plays against B_1, the winner (that is, B_3) plays against B_2, the winner (that is, B_2) plays against A, and finally the winner (that is, A) plays against A_1. □

However, in some restricted cases the two notions of winners coincide. For example, when we have 3 candidates, the necessary Schwartz winners from the incomplete profile and from the incomplete majority graph coincide.

THEOREM 2. *Let P be an incomplete profile over 3 candidates. Then $NecS(P) = NecS(M(P))$.*

PROOF. For every candidate A, we can partition the set of candidates in two sets S_1 and S_2, where S_1 contains the candidates that are reachable from A (i.e., every candidate X such that there is a path $A >_m \cdots >_m X$ from A to X in $M(P)$) and S_2 contains the candidates that are not reachable from A. If there are 3 candidates, (say A, B, and C), then there are four possible kinds of majority graphs, depending on who reaches who else:

- $S_1 = \{A\}$ and $S_2 = \{B, C\}$;
- $S_1 = \{A, B\}$ and $S_2 = \{C\}$;
- $S_1 = \{A, C\}$ and $S_2 = \{B\}$;
- $S_1 = \{A, B, C\}$ and $S_2 = \emptyset$;

In the first case, there is no path from A to B. Therefore, by Theorem 8 of [5], $A \notin NecS(M(P))$. Moreover, B or C has no ingoing edges in $M(P)$. Assume that B has no ingoing edges. Let us consider the completion of P, say P', where we put $B > A$ if the relation between A and B is unspecified, and $B > C$ if the relation between B and C is unspecified. Then, B has only outgoing edges in $M(P')$ and so B is a Condorcet winner, i.e., he wins in every tree. Since there is a completion of P where A is a loser for every tree, $A \notin NecS(P)$.

In the second case, there is no path from A to C. Therefore, by Theorem 8 of [5], $A \notin NecS(M(P))$. Moreover, C has no ingoing edges in $M(P)$. Let us consider the completion of P, say P', where we put $C > A$ if the relation between A and C is unspecified, and $C > B$ if the relation between B and C is unspecified. Then, C has only outgoing edges in $M(P')$ and so B is a Condorcet winner, i.e., he wins in every tree. Since there is a completion of P where A is a loser for every tree, $A \notin NecS(P)$.

In the third case, there is no path from A to B. Therefore, by Theorem 8 of [5], $A \notin NecS(M(P))$. Moreover, B has no ingoing edges in $M(P)$. We can conclude that $A \notin NecS(P)$ via a reasoning that is similar to the one used in the case above.

In the fourth case, there is a path from A to every other candidate. Therefore, $A \in NecS(M(P))$ and so $A \in NecS(P)$ since $NecS(M(P)) \subseteq NecS(P)$. □

The equality between $NecS(P)$ and $NecS(M(P))$ holds also when we have more than 3 candidates, if we impose some other restrictions.

THEOREM 3. *Let P be an incomplete profile. Then $NecS(P) = NecS(M(P))$ if*

- *$M(P)$ is complete, or*
- *$M(P)$ is fully incomplete, or*
- *there are two candidates with no ingoing edges in $M(P)$ (in which case $NecS(P) = NecS(M(P)) = \emptyset$).*

PROOF.

- If $M(P)$ is complete, then $M(P)$ is also the majority graph of every completion of P. Therefore, if $A \notin NecS(M(P))$, there is no path from A to some candidate B in $M(P)$ and in $M(P')$, for every completion P' of P. Therefore, $A \notin NecS(P)$ and so $NecS(P) \subseteq NecS(M(P))$. We can thus conclude that $NecS(P) = NecS(M(P))$, since [8] shows that $NecS(P) \supseteq NecS(M(P))$.
- If $M(P)$ is fully incomplete, then $NecS(M(P)) = \emptyset$. Moreover, if $M(P)$ is fully incomplete, there are two candidates, say B_1 and B_2, with no ingoing edges. If we consider the completion P_1 of P where we put $B_1 > C$ for every C such that the relation between B_1 and C is unspecified, B_1 is a Condorcet winner, i.e., B_1 wins in every tree. Similarly, if consider the completion P_2 of P where we put $B_2 > C$ for every C such that the relation between B_2 and C is unspecified, B_2 is a Condorcet winner, i.e., B_2 wins in every tree. Since in P_1 B_1 wins for every tree, and since in P_2 B_2 wins for every tree, it is not possible to find a unique candidate that in both completions P_1 and P_2 wins for some tree. Therefore, $NecS(P) = \emptyset$.
- If there are two candidates with no ingoing edges in $M(P)$, then we can conclude as in the case above that $NecS(P) = NecS(M(P))$. □

We now consider cases where a candidate is neither a necessary Schwartz winner from the majority graph nor a necessary Schwartz winner from the profile.

THEOREM 4. *Let P be an incomplete profile.*

- *If there is a unique candidate B with no ingoing edges, then, for every other candidate A, $A \notin NecS(M(P))$ and $A \notin NecS(P)$.*
- *For every candidate A, such that there is at least a candidate that A does not reach and all the candidates that are reachable from A in $M(P)$ are beaten by all the other candidates that are not reachable from A, $A \notin NecS(M(P))$ and $A \notin NecS(P)$.*
- *If there is a partition of the candidates in two sets, say S_1 and S_2, such that every element in S_1 is worse than every element in S_2 in $M(P)$, then for every $A \in S_1$, $A \notin NecS(M(P))$ and $A \notin NecS(P)$.*

PROOF.

- If there is a unique candidate B with no ingoing edges, then, for every other candidate $A \neq B$, A does not reach B. Thus, by Theorem 8 of [5], $A \notin NecS(M(P))$. Let us consider the completion P' where we put $B > C$ for every C such that the relation between B and C is unspecified. Then B is a Condorcet winner, i.e., B wins in every tree, and so every other candidate A is a loser for every tree. Therefore $A \notin NecS(P)$.
- Let us consider a candidate A such that all the candidates that are reachable from A in $M(P)$ are beaten by all the other candidates that are not reachable from A. Let us denote with B the candidate that A does not reach. Since A does not reach B, then, by Theorem 8 of [5], $A \notin NecS(M(P))$. Let us consider the completion P_1 of P where we put C above A for every C where the relation between C and A is unspecified in $M(P)$ (and thus also $B > A$ if the relation between A and B is unspecified in $M(P)$) and where we put all the remaining unspecified preferences in an arbitrary way that satisfies transitivity. A cannot reach B in $M(P_1)$. In fact, A cannot reach B directly by construction. Moreover, A cannot reach B via the candidates that A reaches since, by hypothesis, such candidates are all beaten by every candidate that A does not reach in $M(P)$. Therefore, they are all beaten by B in $M(P)$ and thus also in $M(P_1)$. Since A cannot reach B in the complete majority graph $M(P_1)$, by Theorem 8 of [5], A is not a Schwartz winner for P_1, i.e., for P_1, A is a loser for every tree. Therefore, $A \notin NecS(P)$.
- If there is a partition of the candidates in two sets, say S_1 and S_2, such that every element in S_1 is worse than every element in S_2 in $M(P)$, then we can conclude by using a reasoning similar to the one considered in the previous item. □

3.2 Possible winners

Given an incomplete profile P and a simple voting tree T, the possible winners of T for P and for $M(P)$ may be different.

THEOREM 5. *There is an incomplete profile P and a simple voting tree T such that $Poss(P,T) \neq Poss(M(P),T)$.*

PROOF. We can consider the incomplete profile P with just one agent and $\Omega = \{A, B, C\}$, where only the relation between A and B is specified and it is $A > B$. The induced majority graph $M(P)$ has only one edge from A to B. Let us consider the simple voting tree T where A plays against C and the winner plays against B. It is easy to see that $B \in Poss(M(P),T)$, since there is a completion of $M(P)$ where B wins in T, i.e., $B >_m C >_m A$. However, for every completion of P, B does not win in T and so $B \notin Poss(P,T)$. □

3.3 Necessary winners

In general, if there is a necessary winner from an incomplete majority graph $M(P)$, then it is also a necessary winner from the incomplete profile P. More precisely, let P be an incomplete profile and T a simple voting tree, then $Nec(M(P),T) \subseteq Nec(P,T)$.

We will now show that the opposite does not hold in general.

THEOREM 6. *There is an incomplete profile P and a simple voting tree T such that $Nec(P,T) \neq Nec(M(P),T)$.*

PROOF. To show the result, we give an incomplete profile P, a simple voting tree T, and a candidate A such that $A \in Nec(P,T)$ and $A \notin Nec(M(P),T)$.

Let $\Omega = \{A, B, C, D, E, F\}$. Assume to have 5 agents and that the incomplete profile P is defined as follows:

- agent 1: $(E > B > C,\ F > D > A)$;
- agent 2: $(A > E > F > D > B > C)$;
- agent 3: $(A > C > D > F > E > B)$;
- agent 4: $(C > D > F > E > B > A)$;
- agent 5: $(B > A > F > E > C > D)$.

The majority graph of P has the following edges:

- $A >_m C$, $A >_m D$, $A >_m E$, $A >_m F$,
- $B >_m C$, $B <_m D$, $B <_m E$, $B <_m F$,
- $C >_m D$, $C <_m E$, $D <_m F$, $E <_m F$.

Assume that the voting tree T is defined as follows: the winner between C and F plays against the winner between E and D, the winner then plays against B and finally the winner plays against A.

If we apply Algorithm *Win* (which is described in Section 2.5) to T and $M(P)$, then the returned set is $\{A, B\}$. Since T is a simple voting tree, the set $\{A, B\}$ coincides with the set of the possible winners for $M(P)$ and T. Since there are two possible winners, then $Nec(M(P),T) = \emptyset$.

However, $A \in Nec(P,T)$. In fact, for every completion of P, A wins in T. Note that in P only the first agent expresses incomplete preferences. Therefore, the completions of P correspond to the completions of the first agent's preferences. Such completions can be partitioned into two types: those where the first agent puts E above F and those where the first agent puts F above E.

- In every completion of P where E is above F in the first agent's preferences, we have, by transitivity, $E > D$ for this agent, and so there is a majority of agents stating $E > D$. Therefore, for every completion of this kind, its majority graph has the edge $E >_m D$, and thus A wins in T, since C is beaten by F or by E, then B is beaten by F and E, and finally A beats both E and F. Therefore, A is the overall winner.

- In every completion of P where F is above E in the first agent's preferences, we have, by transitivity, $F > C$ for this agent, and so there is a majority of agents stating $F > C$. Therefore, for every completion of this kind, its corresponding majority graph has the edge $F >_m C$, and thus A wins in T, since C is beaten by F, then F beats both D and E, then F beats B, and finally A beats F. Therefore, A is the overall winner. □

However, in some restricted cases the two notions of winners coincide. For example, when we have 3 candidates, the necessary winners from the incomplete profile and from the incomplete majority graph coincide.

THEOREM 7. *Let P be an incomplete profile over 3 candidates and T a simple voting tree. Then, $Nec(P,T) = Nec(M(P),T)$.*

PROOF. When there are three candidates, say A, B, and C, there is only one kind of tree, where one candidate plays against another candidate, and the winner plays against the remaining one. Let us consider the simple voting tree T where A plays against B and the winner plays against C. Every other simple voting tree can be obtained by T by renaming the candidates. We now show that for every possible kind of incomplete majority graph $M(P)$, $Nec(P,T) = Nec(M(P),T)$. We will say $A?_m B$ when the relation between A and B is missing in $M(P)$.

Assume $A >_m B$. If $A >_m C$ or $A <_m C$, $M(P)$ gives all the information for T and thus $Nec(M(P),T) = Nec(P,T)$. If $A?_m C$, $Poss(M(P),T) = \{A, C\}$ and thus $Nec(M(P),T) = \emptyset$. In the completion of P where we put $A > C$ if the relation between A and C is unspecified, A wins in T, while in the completion of P where we put $C > A$, if the relation between A and C is unspecified, C wins in T. Therefore, since there is no a single candidate that in every completion of P wins in T, $Nec(P,T) = \emptyset$.

Assume $A <_m B$. We can conclude similarly to the previous item.

Assume $A?_m B$.

- If $C >_m A$ and $C >_m B$, then $Nec(M(P),T) = Nec(P,T) = \{C\}$.

- If $C <_m A$ and $C <_m B$, then $Poss(M(P),T) = \{A, B\}$ and thus $Nec(M(P),T) = \emptyset$. In the completion of P where we put $A > B$ (resp., $B > A$) if the relation between A and B is unspecified, A (resp., B) wins in T and thus $Nec(P,T) = \emptyset$.

- If $C <_m A$ and $C >_m B$, then $Poss(M(P),T) = \{A, C\}$ and thus $Nec(M(P),T) = \emptyset$. In the completion of P where we put $A > B$ (resp., $B > A$) if the relation between A and B is unspecified, A (resp., C) wins in T and thus $Nec(P,T) = \emptyset$.

- If $C >_m A$ and $C <_m B$, we can conclude similarly to the previous item.

- If $C >_m B$ and $C?_m A$, then $Poss(M(P),T) = \{C, A\}$ and thus $Nec(M(P),T) = \emptyset$. Moreover, in the completion of P where we put $C > A$ if the relation between A and C is unspecified, C wins in T, while in the completion of P where we put $A > B$ and $A > C$, A wins in T. Therefore, since there is no a single candidate that in every completion of P wins in T, $Nec(P,T) = \emptyset$.

- If $C >_m A$ and $C?_m B$, then we can conclude similarly to the previous item.

- If $C <_m B$ and $C?_m A$, then $Poss(M(P),T) = \{A, B, C\}$ and thus $Nec(M(P),T) = \emptyset$. Moreover, in the completion of P where we put $B > A$, if the relation between A and B is unspecified, B wins in T, while in the completion of P where we put $A > C$ if the relation between A and C is unspecified, and $A > B$ if the relation between A and B is unspecified, A wins in T. Therefore, since there is no single candidate that in every completion of P wins in T, $Nec(P,T) = \emptyset$.

- If $C <_m A$ and $C?_m B$, then we can conclude similarly to the previous item. □

The equality between $Nec(P,T)$ and $Nec(M(P),T)$ holds also when we have more than 3 candidates, if we impose some other restrictions.

THEOREM 8. *Let P be an incomplete profile and T a simple voting tree. Then $Nec(P,T) = Nec(M(P),T)$ if*

- *$M(P)$ is complete, or*

- *$M(P)$ is fully incomplete (in which case $Nec(P,T) = Nec(M(P),T) = \emptyset$).*

PROOF.

- If $M(P)$ is complete, then $M(P)$ is also the majority graph of every completion of P. Therefore, if $A \notin Nec(M(P),T)$, then $A \notin Nec(P,T)$. Therefore, $Nec(P,T) \subseteq Nec(M(P),T)$. We can thus conclude that $Nec(P,T) = Nec(M(P),T)$, since $Nec(P,T) \supseteq Nec(M(P),T)$.

- If $M(P)$ is fully incomplete, then all the candidates are possible winners for $M(P)$ and so $Nec(M(P),T) = \emptyset$. Moreover, if $M(P)$ is fully incomplete, there are two candidates, say B_1 and B_2 with no ingoing edges. We can conclude that $Nec(P,T) = \emptyset$ by using a reasoning similar to the one considered in the proof of the second item of Theorem 3. More precisely, in the completion P_1 of P where we put $B_1 > C$ for every C such that the relation between B_1 and C is unspecified, B_1 is a Condorcet winner, and so B_1 wins also in T, while in the completion P_2 of P where we put $B_2 > C$ for every C such that the relation between B_2 and C is unspecified, B_2 is a Condorcet winner, and so B_2 wins also in T. Therefore, it is not possible to find a unique candidate that in the completions P_1 and P_2 wins in T. Therefore, $Nec(P,T) = \emptyset$. □

We can also identify cases in which a candidate is neither a necessary winner from the majority graph nor a necessary winner from the profile.

THEOREM 9. *Let P be an incomplete profile and T a simple voting tree. For every candidate A such that, for every candidate C that may play against A in T, $A <_m C$ or the relation between A and C is unspecified in $M(P)$, $A \notin Nec(M(P),T)$ and $A \notin Nec(P,T)$.*

PROOF. If, for every candidate C that may play against A in T, we have $A <_m C$, then $A \notin Poss(M(P),T)$. Thus $A \notin Nec(M(P),T)$ and $A \notin Poss(P,T)$. Since $A \notin Poss(P,T)$, we have that $A \notin Nec(P,T)$. If there is a candidate C that may play against A in T, such that the relation between A and C is unspecified in $M(P)$, then either $A \notin Poss(M(P),T)$ (and so we can conclude as above) or $A \in Poss(M(P),T)$ but $|Poss(M(P),T)| > 1$. Then, by Algorithm *StrongWin* [6], $A \notin Nec(M(P),T)$. Moreover, let us consider the completion of P where we put $A < C$ for every C such that the relation between A and C is unspecified in $M(P)$. Then, for every tree (and thus also in T) A is a loser. Therefore, $A \notin Nec(P,T)$. □

4. WINNERS IN VOTING TREES

We now close the open questions of Table 1 in the voting tree column. We will show that algorithms *Win* and *StrongWin* [6], that correctly compute possible and necessary winners from an incomplete majority graph and a simple voting tree, are not correct any longer when we consider voting trees.

4.1 Winners

We first notice that, since a simple voting tree is a voting tree, all the inequality results shown in the previous section (or already known and shown in Table 1) for simple voting trees hold also for voting trees. Thus, we have inequalities for the notions of possible Schwartz winners, necessary Schwartz winners, possible winners, and necessary winners.

For the remaining notions of possible and necessary Condorcet winners, we will now show that the equalities that hold for simple voting trees hold also for voting trees.

THEOREM 10. *Let P be an incomplete profile. Assume we consider voting trees. Then*

- $PossCond(P) = PossCond(M(P))$;
- $NecCond(P) = NecCond(M(P))$.

PROOF. It follows from the proofs of items 3 and 4 of Theorem 1 of [8], which show that these equalities hold for simple voting trees, since this proof depends only on the completions of $M(P)$ and of P and not on the kind of voting trees considered. □

4.2 Majority graph winners

It is easy to see that all the polynomial time algorithms presented for incomplete majority graphs and simple voting trees in [5] are sound and complete also for voting trees, since they don't exploit the fact that each candidate appears exactly once in the leaves.

However, we can show that, when we consider voting trees instead of simple voting trees, algorithms *Win* and *StrongWin* [6], that compute possible and necessary winners from an incomplete majority graph and a simple voting tree, are not correct. In fact, given a voting tree T and a majority graph M, Algorithm *Win* may return also candidates that are not possible winners for M and T, and algorithm *StrongWin* may return the empty set even if there is a necessary winner for M and T.

THEOREM 11. *There is an incomplete majority graph M and a voting tree T such that, if W is the set of candidates returned by Algorithm* Win [6] *applied to M and T, then $W \neq Poss(M,T)$.*

PROOF. We give an incomplete majority graph M and a voting tree T, such that the set of candidates returned by Algorithm *Win* [6] applied to T and M contains also a candidate which is not a possible winner for T and M.

Assume we have 5 candidates, say A, B, C, D, and E, and that the incomplete majority graph M has the following edges: $A >_m D$, $A >_m E$, $B >_m A$, $B >_m E$, $C >_m A$, $C >_m D$, $D >_m B$, $E >_m C$. Assume that the voting tree T is defined as follows:

- $left(root(T))$ is the voting tree where first B plays against C, the winner plays against D, and finally the winner plays against A, and
- $right(root(T))$ is the voting tree where first B plays against C, the winner plays against E, and finally the winner plays against A.

The set of the candidates returned by Algorithm *Win* [6] is the set $\{A,B,C\}$, while the set of possible winners for M and T is the set $\{B,C\}$. A is not a possible winner for M and T, since there are only two completions of M, i.e., the one, say c_1, where we put $B > C$ and the one, say c_2, where we put $C > B$. In the first one, the winner is B, while in the second one the winner is C:

- In completion c_1, A is the winner of $left(root(T))$ (since B beats C, then B is beaten by D, and finally D is beaten by A), B is the winner of $right(root(T))$ (since B beats C, then B beats E, and finally B beats by A), and thus, since $B >_m A$ in M, B is the winner of T.
- In completion c_2, C is the winner of $left(root(T))$ (since C beats B, then C beats D, and finally C beats A), A is the winner of $right(root(T))$ (since C beats B, then C is beaten by E, and finally E is beaten by A), and thus, since $C >_m A$ in M, C is the winner of T. □

THEOREM 12. *There is an incomplete majority graph M and a voting tree T such that, if W the set of candidates returned by Algorithm* StrongWin [6] *applied to M and T, $W \neq Nec(M,T)$.*

PROOF. We give an incomplete majority graph, say M', and a voting tree, say T', such that the set of the candidates returned by Algorithm *StrongWin* [6] applied to T' and M' is empty, while there is a candidate that is a necessary winner for T' and M'.

Assume we have 6 candidates, say A, B, C, D, E, and F, and that the incomplete majority graph M' has the same edges as the incomplete majority graph M considered in the proof of Theorem 11 plus the following edges: $A >_m F$, $F >_m B$, and $F >_m C$. Assume that the voting tree T' is

the voting tree where the winner of the voting tree T considered in the proof of Theorem 11 plays against F. Algorithm *StrongWin* applied to M' and T' returns the empty set, since Algorithm *Win* returns a set (i.e., $\{A, F\}$) which has more than one element. However, we have shown in the proof of Theorem 11 that, for every completion of M, A does not win in T. Thus, by construction of M', for every completion of M', A does not win in T, and so, since $F >_m B$ and $F >_m C$ in M', for every completion of M', F wins in T. Therefore, F is the necessary winner for M' and T'. Hence, there is a necessary winner for M' and T' but Algorithm *StrongWin* says there is none. □

However, even if algorithm *Win* may be incorrect, the set of winners returned by this algorithm may be useful in the search for the possible/necessary winners. In fact, this is a superset of the set of the possible winners for M and T. Moreover, if Algorithm *StrongWin* applied to M and T returns a candidate, this is indeed a necessary winner for M and T.

THEOREM 13. *Let M be an incomplete majority graph, T a voting tree, and W the set of the candidates returned by Algorithm* Win [6]. *Then $W \supseteq Poss(M,T)$.*

PROOF. Let A be a candidate. We want to show that, if $A \in Poss(M,T)$, then $A \in W$. Assume that $A \notin W$. Then, by definition of Algorithm *Win*, for every subtree T' of T where $A \in root(left(T'))$ (resp., $A \in root(right(T'))$, we have $A <_m C$, for every $C \in root(right(T'))$ (resp., for every $C \in root(left(T'))$). This holds for every completion of M. Therefore, for every completion of M, A is a loser in T and thus $A \notin Poss(M,T)$. □

COROLLARY 13.1. *Let M be an incomplete majority graph, T a voting tree, and W the set of the candidates returned by Algorithm* Win [6]. *If $|W| = 1$, then $W = Nec(M,T)$.*

PROOF. If $W = \{A\}$, then, by Theorem 13 and by the fact that $Poss(M,T)$ cannot be empty, we have that $Poss(M,T) = \{A\}$ and thus $Nec(M,T) = \{A\}$. □

5. CONCLUSIONS

In the setting of voting trees with missing preferences and/or uncertain agenda, we have closed all open questions about the relation between the notions of winners for the profile (which are the winners we ultimately want to compute) and the corresponding notions for the majority graph (which are polynomial to find). More precisely, the overall results are summarized in Table 2, where we have inserted our new results in those of Table 1, which described the previous state of the art.

	SVT	VT
Possible Schwartz winners	$\neq$ [8]	$\neq$
Necessary Schwartz winners	$\neq$	$\neq$
Possible Condorcet	$=$ [8]	$=$
Necessary Condorcet	$=$ [8]	$=$
Possible winners	$\neq$	$\neq$
Necessary winners	$\neq$	$\neq$

Table 2: New state of the art about winners computed from the majority graph or profile.

Where we see an equality in the table, it means that we can safely and correctly work from the majority graph. This is both more compact and efficient. This happens for possible/necessary Condorcet winners. Instead, where we see an inequality, it means that using the majority graph may give us an upper or lower approximation of the desired set of winners. However, this approximation can be found in polynomial time. The approximation is closer to the correct notion in the case of simple voting trees, since algorithms *Win* and *StrongWin* are more correct for simple voting trees than for voting trees.

6. ACKNOWLEDGMENTS

This work has been partially supported by the MIUR PRIN 20089M932N project "Innovative and multi-disciplinary approaches for constraint and preference reasoning". We would like to thank Jerome Lang for many stimulating discussions.

7. REFERENCES

[1] V. Conitzer, T. Sandholm, and J. Lang. When are elections with few candidates hard to manipulate. *Journal of the ACM*, 54(3), 2007.

[2] R. Farquharson. *Theory of voting*. Yale University Press, New Haven, 1969.

[3] M. JoséHerrero and S. Srivastava. Implementation via backward induction. *Journal of Economic Theory*, 56(1):70 – 88, 1992.

[4] K. Konczak and J. Lang. Voting procedures with incomplete preferences. In *Proceedings of IJCAI'05 Multidisciplinary Workshop on Advances in Preference Handling*, 2005.

[5] J. Lang, M. S. Pini, F. Rossi, K. B. Venable, and T. Walsh. Winner determination in sequential majority voting. In *Proceedings of IJCAI'07*, pages 1372–1377. AAAI Press, 2007.

[6] M. S. Pini, F. Rossi, K. B. Venable, and T. Walsh. Determining winners in weighted sequential majority voting: incomplete profiles w.r.t. majority graphs. In *Proceedings of CLIMA VIII*, 2007.

[7] M. S. Pini, F. Rossi, K. B. Venable, and T. Walsh. Incompleteness and incomparability in preference aggregation. In *Proceedings of IJCAI'07*, pages 1464–1469. AAAI Press, 2007.

[8] M. S. Pini, F. Rossi, K. B. Venable, and T. Walsh. Dealing with incomplete agents' preferences and an uncertain agenda in group decision making via sequential majority voting. In *Proceedings of KR'08*, pages 571–578. AAAI Press, 2008.

[9] A. D. Procaccia, A. Zohar, Y. Peleg, and J. S. Rosenschein. Learning voting trees. In *Proceedings of AAAI*, pages 110–115, 2007.

[10] M. Trick. Small binary voting trees. In *Proceedings of COMSOC'06*, pages 500–511, 2006.

[11] V. Vassilevska. Fixing a tournament. In *Proceedings of AAAI'10*, 2010.

[12] T. Walsh. Complexity of terminating preference elicitation. In *Proceedings of AAMAS'08*, pages 967–974, 2008.

[13] L. Xia and V. Conitzer. Determining possible and necessary winners under common voting rules given partial orders. In *Proceedings of AAAI'08*, pages 196–201. AAAI Press, 2008.