# Winner Determination in Sequential Majority Voting with Incompleteness*

J. Lang
LAMSADE

M. S. Pini
University of Padova

F. Rossi
University of Padova

K. B. Venable
University of Padova

T. Walsh
NICTA and UNSW

**Abstract**

In multiagent settings where the agents have different preferences, preference aggregation is a central issue. Voting is a general method for preference aggregation. We consider the sequential majority voting (SMV) rules to aggregate agents' preferences. According to these rules each decision is taken by performing a sequence of pairwise comparisons along a binary tree where each comparison is a weighted majority vote among the agents. Incompleteness in the agents' preferences is common in many real-life settings due to privacy issues or an ongoing elicitation process. We study how to determine winners when preferences may be incomplete, not only for the SMV rule (where the tree is assumed to be fixed), but for two other closely related rules: the *Schwartz* rule, for which the winners are the candidates winning the SMV rule for at least one tree, and a new rule, that we call the *fair SMV rule*, for which the winners are the candidates winning the SMV rule for at least one *balanced* tree. For each of these rules, we address the complexity of computing necessary (resp. possible) winners, which are respectively those candidates winning for all completions (resp. at least one completion) of the incomplete profile. For the fair SMV rule, we also address the complexity of computing the winner when the profile is complete. Most of these problems are computationally hard (in some cases, the exact complexity is still unknown). Since it is generally difficult to find the exact set of winners for the SMV, Schwartz and fair SMV rules, we propose several polynomial algorithms that find approximations of the sets of possible and necessary winners, based on the completions of the (incomplete) majority graph built from the incomplete profiles.

## 1 Introduction

In multiagent settings, agents generally have different preferences, and it is of central importance to be able to aggregate these preferences, i.e. to select a collectively desirable candidate from a set of candidates. Candidates could be, for example, potential

*This paper is a revised and extended version of the conference papers [10, 15].

presidents, joint plans, allocations of goods or resources, etc. A general method for aggregating preferences in multi-agent systems, in order to take a collective decision, is running an election among the different options using a voting rule.

Unfortunately, eliciting preferences from agents, to be able to run such an election, that is a typical problem in multiagent systems such as combinatorial auctions and voting systems, is a difficult, time-consuming and costly process. The set of alternatives that agents have to consider to give their preferences is often too large, hence they usually gives only part of their preferences. Moreover, sometimes agents may also be unwilling to reveal all their preferences for privacy reasons. Fortunately, we can often determine the outcome before all the preferences have been revealed [4]. For example, it may be that one option has so many votes that it will win whatever happens with the remaining votes.

We consider the class of sequential majority voting (SMV) rules (sometimes called the "Cup" or "tournament" rules), which has been extensively studied in Social Choice Theory (*e.g.*, [12, 11]), where preferences are aggregated by a sequence of pairwise comparisons along a binary tree. We then consider the Schwartz rule, which can be characterized as a SMV rule with an undefined tree, as the Schwartz winners are exactly those who win the SMV rule for at least one competition tree. In other words, a candidate is a Schwartz winner if the chair can see to it that it is a winner by fixing the tree. Lastly, we define a new voting rule strongly related to SMV rules: the *fair SMV* rule elects candidates who win the SMV rule for at least one *balanced tree*. Fairness comes from the fact that whatever the finalists, they will have faced the same number of competitions, or the same number plus or minus one. While computing the winner for any SMV rule as well as for the Schwartz rule is obviously polynomial, the fair SMV rule appears to be much more complex. We partially assess the complexity of the fair SMV rule, showing that in the presence of *weighted* votes, computing the winner is NP-hard (the exact complexity of the fair SMV rule is still an open problem).

Then we turn to the computation of the above rules (SMV for a fixed tree, Schwartz, fair SMV) when voters' preferences are incomplete, *i.e.*, represented by partial orders on the set of candidates. We focus on the computation of *necessary winners* (candidates who win for any completion of these partial preferences) and *possible winners* (candidates who win for some completion of these partial preferences). Computing necessary and possible winners in SMV for a fixed tree, Schwartz and fair SMV are likely to be hard, and we show that they are indeed when the input consists of weighted votes. Now, we introduce an approximation of these possible and necessary winners, where the set of completions of the incomplete preferences is replaced by the completions of the *incomplete majority graph* computed from the incomplete preferences. We discuss the links between possible (resp. necessary) winners and their approximation from the incomplete majority graph, and we show that these approximations can be computed in polynomial time.

## 2 Sequential majority voting

We now give some basic notions on sequential majority voting. First, we focus on unweighted scenarios and then we redefine all the notions in the context of weighted votes.

## 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, irreflexive and transitive order) on a set of candidates. The candidates, are taken from a set $\Omega$ and they represent the possible options over which agents will vote.

**Definition 1 (profile)** *A* profile $P$ *on* $\Omega$ *is a collection of* $n$ linear *orders over* $\Omega$*, i.e.,* $P = (P_1, \ldots, P_n)$*, where* $P_i$ *is the preference relation (or* vote*) of agent* $i$*.*

For the sake of simplicity we assume throughout the paper that the number $n$ of voters is odd. The notions results we give in this paper can easily extended to the case where $n$ is even, but at the price of some complications that arise with the handling of ties.

Profiles are denoted using the following notation: (($A > B > C$); ($A > C > B$); ($C > A > B$)) means that voter 1 prefers $A$ to $B$ and $B$ to $C$, etc.

**Definition 2 (voting rule)** *A (deterministic) voting rule is a mapping from the set of profiles to the set of candidates. A voting correspondence is a mapping from the set of profiles to the set of nonempty sets of candidates.*

For the sake of simplicity, we will use the word "rule" even for correspondences. This will not cause any ambiguity.

Given a profile $P$, the induced majority graph $M(P)$ is defined as follows.

**Definition 3 (majority graph)** *Let* $P$ *be a profile. The majority graph induced by* $P$*, denoted by* $M(P)$*, is the graph whose set of vertices is the set of candidates* $\Omega$ *and where, for all* $A, B \in \Omega$*, there is a directed edge from* $A$ *to* $B$ *(denoted by* $A >_m^P B$ *iff a strict majority of voters prefer* $A$ *to* $B$*.*

The majority graph is asymmetric and irreflexive, but it is not necessarily transitive. Moreover, since the number of voters is odd, $M(P)$ is complete: for each $A$, $B \neq A$, either $A >_m B$ or $B >_m A$ holds. Therefore, $M(P)$ is a complete, irreflexive and asymmetric graph, also called a *tournament* on $\Omega$ [11].

**Definition 4 (weighted majority graph)** *Given a profile* $P$*, the weighted majority graph induced by* $P$ *is the graph* $M_W(P)$ *whose set of vertices is* $\Omega$ *and in which for all* $A, B \in \Omega$*, there is a directed edge from* $A$ *to* $B$ *weighted by the number of voters who prefer* $A$ *to* $B$ *in* $P$*.*

## 2.2 Sequential majority voting with a fixed tree

Given a set of candidates, the sequential majority voting rule is defined by a binary tree (also called an *agenda*) with one candidate per leaf, such that each candidate appears on exactly one leaf. [1]Each internal node represents the candidate that wins the pairwise election between the node's children. The winner of a pairwise election is computed by the majority rule, where $A$ beats $B$ iff there is a majority of voters stating $A > B$. The candidate at the root of the agenda is the overall winner. Given a binary tree $T$, we will denote with $SMV(T)$ the sequential majority voting rule defined by $T$, and given a profile $P$, we will denote with $SMV(P,T)$ the set of the candidates that, given profile $P$, win in tree $T$. Note that, because the number of voters is odd, ties never happen in pairwise elections, so that $SMV(T)$ is a (deterministic) voting rule. When the binary tree is unknown, we will denote with $SMV$ the class of sequential majority voting rules.

**Example 1** *Let $\Omega = \{A, B, C, D\}$. Consider the agenda $T$ shown in Figure 1 (a), where $A$ is first confronted against $D$, the winner, called $w_1$, is confronted against $C$, and then the winner, called $w_2$, is confronted against $B$. The winner, called $w_3$, is the overall winner. Consider the profile $P = ((A > B > C > D); (D > A > B > C); (C > D > B > A))$. Its induced majority graph is shown in Figure 1 (b). Therefore $SMV(P,T) = B$.* □

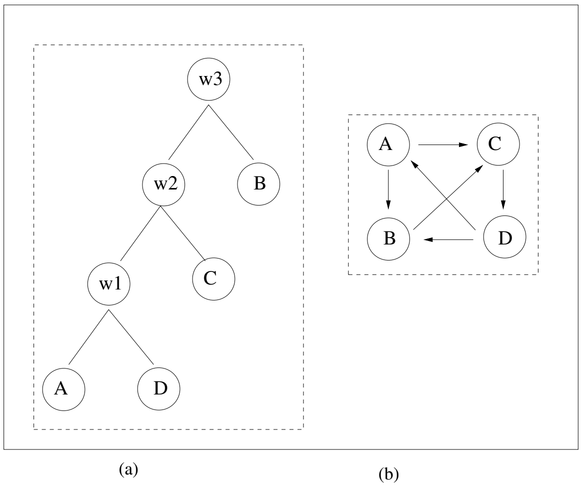


Figure 1: An agenda and a tournament.

## 2.3 The Schwartz rule

Given a profile $P$, a Schwartz winner [16] is a candidate that wins, according to the preferences of $P$, in some agenda.

[1]More general families of sequential majority rules do not make this assumption. More of our results would remain true if some candidate were allowed to appear at several leaves, or none.

**Definition 5 (Schwartz winner)** *Given a profile $P$, a candidate $A$ is a Schwartz winner iff there exists a binary tree $T$ such that $SMV(P,T) = A$.*

The Schwartz rule (more precisely, correspondence) maps any profile $P$ to the set of all Schwartz winners for $P$. The set of Schwartz winners coincides with the *top cycle* of the majority graph [12]. The top cycle of a majority graph $G$ is the set of maximal elements of the reflexive and transitive closure $G^*$ of $G$.

**Example 2** *Consider the profile presented in Example 1, i.e., $P = ((A > B > C > D); (D > A > B > C); (C > D > B > A))$. Then the set of Schwartz winners for $P$ is $\{A, B, C, D\}$.* □

Yet an equivalent characterization of Schwartz winners, that we use further in the paper, is the following:

**Theorem 1 (see e.g. [12, 11])** *Given a majority graph $G$, a candidate $A$ is a Schwartz winner iff, for every other candidate $C$, there exists a path from $B_i$ to $B_{i+1}$ for $i = 1, \ldots, k-1$ where $B_1 = A$ and $B_k = C$.*

Obviously, from either of these two characterizations, computing the Schwartz winners is polynomial.

Note that sequential majority, as well as Schwartz, are among the rules that can be computed from the majority graph. Among the most well-known rules computable from the majority graph, the Schwartz rule is the most liberal: at the other extremity of the spectrum, the Condorcet set contains the Condorcet winner when there exists one, and is empty otherwise; we recall that a Condorcet winner is a candidate who beats every other candidate in the majority graph. In particular, the Schwartz set contains the uncovered set, the Banks set, the set of Copeland winners, the set of Slater winners, and the Condorcet winner when there exists one. Note that all these rules are Condorcet-consistent: when a Condorcet winner $A$ exists, all these sets coincide with the singleton $\{A\}$.

## 2.4 Fair sequential majority voting

Schwartz winners are candidates who win in at least one binary tree. However, such a tree may be very unbalanced and the winner could compete with few other candidates. This might be considered "unfair". In the following, we will consider a competition *fair* if it has a balanced agenda.

**Definition 6 (balanced agenda)** *A balanced agenda $T$ is a binary tree in which each candidate must appear exactly once in leaves and the difference between the maximum and the minimum depth among the leaves is less than or equal to 1.*

Assume to have complete information about preferences. We will call fair winners the winners of a balanced agenda.

**Definition 7 (fair winner)** *Given a profile $P$, a candidate $A$ is a fair winner iff there exists a balanced agenda $T$, such that $SMV(P,T) = A$. Fair sequential majority voting (FSMV) is the voting rule returning all fair winners.*

Notice that FSMV is a new voting rule (more precisely, correspondence) that has not been considered before, as far as we know.

**Example 3** *Consider the profile presented in Example 1, i.e.,* $P = ((A > B > C > D); (D > A > B > C); (C > D > B > A))$*. The set of the fair winners for* $P$ *is* $\{A, D\}$*. In fact, we have only three balanced trees, up to isomorphism: in the balanced tree where* $A$ *is confronted against* $B$*, and* $C$ *is confronted against* $D$*, the winner is* $A$*, while in the other two balanced trees the winner is* $D$*.* □

Notice that a Condorcet winner is a fair winner, since it wins in all trees, thus also in balanced ones. Moreover, a fair winner is a Schwartz winner, since there is a balanced tree, and so there is a tree, where he wins. Moreover, while there could be no Condorcet winner, there is always at least one fair winner, because there always exist at least a balanced tree. Notice also that the candidates that win in every balanced tree are Condorcet winners.

## 2.5 Weighted votes

Until now we have considered unweighted votes. However, weighted voting systems are used in a number of real-world settings, like shareholder meetings and elected assemblies. An agent with weight an integer $k$ can be viewed as $k$ agents voting identically. Although human elections are often unweighted, the addition of weights make voting schemes more general. Weights are useful in multiagent systems where we have different types of agents and some agents are more important than others when taking a decision. Moreover, the weighted case informs us about unweighted case when we have probabilistic information about votes [3]. For example, in [3] they show that, if coalitional manipulation is hard in the weighted complete-information case, evaluating a candidate's winning probability is hard even in the unweighted case when there is uncertainty about the votes

We now generalize the notions presented above in this section assuming that the agents may have different weights. First, we give the definition of weighted profile and, given a weighted profile, we show how to compute its corresponding unweighted profile. Next, we show how to deal with weights in the majority graphs and in sequential majority voting.

**Definition 8 (weighted profile)** *A weighted profile is a collection of* $n$ *strict total orders describing the preferences for* $n$ *agents, each of which has a given weight.*

Note that the unweighted case considered previously is a special case of weighted case, where all the agents' weights are equal to 1.

For simplicity, we assume that the sum of the weights of the agents is odd. The assumption to have an odd sum of weights ensures that there is never a tied result. This simplification is not essential. We can have an even sum of weights, but in this case we have to specify how we deal with tied results. Thus, for simplicity we assume that the sum of weights is odd.

**Definition 9 (corresponding unweighted profile)** *Given a weighted profile $P$, its corresponding unweighted profile $U(P)$ is the profile obtained from $P$ by replacing every ordering, say $O$, expressed by an agent with weight $k_i$ by $k_i$ agents with weight $1$ all expressing ordering $O$.*

**Example 4** *Consider the profile shown in Example 1, i.e., $P = ((A > B > C > D);$ $(D > A > B > C); (C > D > B > A))$. Assume that the first agent has weight $1$, and that the second and the third agent have weight $2$. Then the corresponding unweighted profile is $U(P) = ((A > B > C > D); (D > A > B > C); (D > A > B > C); (C > D > B > A); (C > D > B > A))$.* □

**Definition 10 (majority graph from a weighted profile)** *Given a weighted profile, the induced* majority graph *is a graph whose set of vertices is the set of candidates $\Omega$, and, for all $A, B \in \Omega$, there is a directed edge from $A$ to $B$ in $M(P)$ (denoted by $A >_m B$) iff a strict* weighted *majority of voters prefer $A$ to $B$.*

Since the sum of weights is odd, $M(P)$ is complete: for each $A$, $B \neq A$, either $A >_m B$ or $B >_m A$ holds. Therefore, as for unweighted profiles, $M(P)$ is a complete, irreflexive and asymmetric graph.

**Example 5** *Consider the weighted profile of Example 4. The majority graph induced by such a profile is equal to the one shown in Figure 1 (b).* □

Sequential majority voting for weighted profiles works as in the case of unweighted profiles, except that the winner of every pairwise competition is computed by the *weighted* majority rule (and not by the classical majority rule), where $A$ beats $B$ iff there is a weighted majority of votes stating $A > B$.

# 3 From incomplete profiles to incomplete majority graphs

When votes are only partially known, applying a voting rule may result in some uncertainty about the winner. We first define what we mean by an incomplete profile; then we will make clear what we mean by applying a voting rule to an incomplete preference profile, via the definition of possible and necessary winners; finally, in the specific case of the voting rules we focus on in this paper, we define an approximation of these two sets of winners, computed directly from the incomplete majority graph induced from the incomplete profile.

## 3.1 Incomplete profiles, possible and necessary winners

Up till now, agents defined completely their preferences over candidates, we intend to consider when the agents' preferences may only be partially known.

**Definition 11 (incomplete profile)** *Let $\Omega$ be a set of* candidates *and $\{1, \ldots, n\}$ a set of* voters*. An* incomplete preference relation *$>$ on $\Omega$ is a strict order on $\Omega$, that is, a transitive and irreflexive relation on $\Omega$. An ($n$-voter)* incomplete profile *on $\Omega$ is a collection $P = (P_1, \ldots, P_n)$ of incomplete preference relations on $\Omega$.*

**Example 6** *Assume there are three voters and three candidates $A$, $B$, and $C$, then a possible incomplete profile is (($A > B > C$); ($B > A$); ($A > B$)).* $\square$

This definition naturally extends to weighted voters: a weighted incomplete profile is a weighted collection of incomplete preference relations.

**Definition 12 (complete extension (aka completion))** *Let $P = (P_1, \ldots, P_n)$ be an incomplete profile over a set of candidates $\Omega$. A complete extension (or completion) $R$ of $P$ is a tuple $(R_1, \ldots, R_n)$ such that every $R_i$ is a* linear order *on $\Omega$ containing $P_i$. We will denote with $Ext(P)$ the set of all the complete extensions of $P$.*

**Definition 13 (possible and necessary winners)** *Let $r$ be a voting rule, $P$ be an incomplete profile and $A$ a candidate. $A$ is a* possible winner *for $P$ and $r$, denoted by $A \in Poss(P, r)$, if there exists a complete extension $R$ of $P$ such that $r(P) = A$. $A$ is a* necessary winner *for $P$ and $r$, denoted by $A \in Nec(P, r)$, if for any complete extension $R$ of $P$ we have $r(P) = A$.*

Possible and necessary winners were introduced in [9]. Then [19] showed that for most commonly used voting rules, computing possible and necessary winners was NP-hard.

These definitions apply to any voting rule, therefore they apply to SMV($T$), Schwartz and fair SMV. For instance, $A$ is a *possible Schwartz winner* for $P$ if it is a possible winner w.r.t. the incomplete profile $P$ and the Schwartz rule $r_{Schwartz}$, i.e., $A \in Poss(P, r_{Schwartz})$.

We also define possible and necessary Condorcet winners [9]:

**Definition 14** *Let $P$ be a profile and $A$ a candidate, $A$ is a* possible Condorcet winner *for $P$ if there exists a completion $Q$ of $P$ such that $A$ is a Condorcet winner for $Q$, and a* necessary Condorcet winner *for $P$ if for every completion $Q$ of $P$, $A$ is a Condorcet winner for $Q$.*

## 3.2 Winners defined from the majority graph

**Definition 15 (incomplete majority graph)** *Given an incomplete (weighted) profile $P$, the incomplete majority graph $M(P)$ induced by $P$ is the graph whose set of vertices is $\Omega$ and where there is an edge from $A$ to $B$, denoted by $A >_m^P B$, if the number of voters (or the sum of the weights of the voters) who prefer $A$ to $B$ is greater than $n/2$.*

$M(P)$ is an incomplete asymmetric graph (or *partial tournament*) over $\Omega$.

**Example 7** *Consider the incomplete profile shown in Example 6, i.e., (($A > B > C$); ($B > A$); ($A > B$)), and assume that the weight of the first agent is $1$ an the weight of every other agent is $2$. The majority graph induced by this incomplete weighted profile, that we call $Q$, is the graph with three nodes $A$, $B$ and $C$ and only one edge $A >_m^Q B$.* $\square$

A *completion* of $M(P)$ is a (complete) tournament containing $M(P)$. Note that the set of all completions of $M(P)$ is a *superset* of the set of majority graphs induced by all possible completions of $P$. Formally, let $Ext(M(P))$ be the set of all complete extensions of $M(P)$ and $M(Ext(P)) = \{M(R) \mid R \in Ext(P)\}$ the set of all majority graphs induced from the extensions of $P$. Then:

**Proposition 1** $M(Ext(P)) \subseteq Ext((M(P))$.

**Proof:** From its definition, we have that $G \in Ext(M(P))$ if and only if for all $(A, B)$ such that $A >_m^P B$, $(A, B) \in G$. Now, let $G \in M(Ext(P))$, which means there is a $Q \in Ext(P)$ such that $G = M_Q$. Now, $A >_m^P B$ clearly implies $A >_m^Q B$, therefore, $M(Q) \in Ext(M(P))$.

This inclusion can be strict, which means that moving from $P$ to $M(P)$ implies a loss of information. An incomplete majority graph throws away information about how individual agents have voted. For example, we lose information about correlations between votes. Such correlations may prevent a candidate from being able to win.

**Example 8** *Consider an incomplete profile $P$ with just one agent and $\Omega = \{A, B, C\}$, where the only vote is $A > B$ (the relations between A and C, and between B and C have not been specified). The induced majority graph $M(P)$ has only one edge from $A$ to $B$. In this situation, there is a completion of $M(P)$ (with edges from $B$ to $C$ and from $C$ to $A$) and an agenda where $B$ wins (we first compare $A$ with $C$, $C$ wins, and then $C$ with $B$, and $B$ wins). However, there is no way to complete profile $P$ and set up the agenda so that B wins. In fact, the possible completions of $P$ are $A > B > C$, $A > C > B$, and $C > A > B$, and in all these cases $B$ is always beaten at least by $A$. Hence, there is no agenda where $B$ wins. Note that the completion of the majority graph that allows us to conclude that $B$ can win, cannot be obtained in any possible completion of the agent's preferences of $P$, since it violates transitivity.* $\square$

Clearly, the more "complete" $P$, the more complete $Ext(P)$ – $M(P)$ can even be complete without $P$ being so – and the better $M(P)$ approximates $M(Ext(P))$.

Now, for any rule $r$ being based on the majority graph, including SMV($T$), Schwartz and fair SMV, we can define notions of possible (resp. necessary) winner from the incomplete majority graph, which will approximate possible and necessary winners (and may be much easier to compute).

**Definition 16 (possible/necessary winner from the incomplete majority graph)** *Let $P$ be an incomplete profile, $A$ a candidate, and $r$ a voting rule based on the majority graph, that is, there is a function $f_r$ such that $r(P) = f_r(M(P))$. $A$ is a* possible winner *for $M(P)$ and $r$ if there exists a completion $M'$ of $M(P)$ such that $A = f_r(M')$. $A$ is a* necessary winner *for $M(P)$ and $r$ if for every completion $M'$ of $M(P)$ we have $A = f_r(M')$. We denote by $Poss(M(P), r)$ and $Nec(M(P), r)$[2] respectively the set of possible and necessary winners for $M(P)$ and $r$.*

[2]Note that, given a voting rule $r$, $Poss(x, r)$ and $Nec(x, r)$ represent two different functions, depending on whether the first argument $x$ is a profile or a majority graph.

Note that these notions of possible and necessary winners only apply to rules that are based on the majority graph.

Note that $Poss(M(P), r) \subseteq Nec(M(P), r)$, and that when the majority graph is complete, $Poss(M(P), r) = Nec(M(P), r) = r(P)$.

We also define possible and necessary Condorcet winners from a majority graph in a similar way:

**Definition 17** *Let $P$ be a profile and $A$ a candidate, $A$ is a* possible Condorcet winner *for $M(P)$ if there exists an extension $M'$ of $M(P)$ such that $A$ beats every other candidate in $M'$. $A$ is a* necessary Condorcet winner *for $M(P)$ if for every extension $M'$ of $M(P)$, $A$ beats every other candidate in $M'$.*

We will denote with $PossCond(P)$ and $NecCond(P)$ (resp., $PossCond(M(P))$ and $NecCond(M(P))$) the possible and necessary Condorcet winners from the profile $P$ (resp., from the majority graph $M(P)$).

**Example 9** *Consider the profile $P$ given in Example 7. We have that $NecCond(P) = Nec(P, r_{Schwartz}) = \emptyset$, $PossCond(P) = \{A, C\}$, and $Poss(P, r_{Schwartz}) = \{A, B, C\}$. Notice that in this example the possible and necessary Schwartz and Condorcet winners obtained by considering the completions of $P$ coincide with those obtained by considering the completions of the majority graph induced by $P$. However, as shown in Example 8, this is not true in general.*

### 3.2.1 Unweighted profiles.

Given an incomplete unweighted profile $P$ and the incomplete majority graph $G$ induced by $P$, that is, $G = M(P)$, we already observed that the completions of $G$ are a (possibly proper) superset of the set of complete majority graphs induced by all possible completions of $P$. This observation leads to the following results.

**Proposition 2** *Let $P$ be an incomplete profile and $r$ be a rule based on the majority graph.*

1. $Poss(M(P), r) \supseteq Poss(P, r)$.
2. $Nec(M(P), r) \subseteq Nec(P, r)$.
3. $PossCond(P) = PossCond(M(P))$.
4. $NecCond(P) = NecCond(M(P))$

**Proof:** Points 1 and 2 are straightforward consequences of Proposition 1.

For point 3, if a candidate $A$ is a possible Condorcet for $P$ then, by Proposition 1, it follows that it is also a possible Condorcet winner for $M(P)$. Also the converse holds. If a candidate $A$ is a possible Condorcet winner for $M(P)$, then there must be one or more completions of the majority graph where $A$ is a Condorcet winner, i.e., where there is an outgoing edge from $A$ to all the other candidates. We now show that there is completion of $P$ in which $A$ is a Condorcet winner. For this we complete $P$ by instantiating all the unspecified relations between $A$ and $C$ with $A > C$ (for all $C$) and all the remaining unspecified preferences in an arbitrary way that preserves transitivity. This is a completion of $P$, say $P'$, since it does not violate transitivity. Moreover, according to Proposition 1 we have, $M(P') \in Ext(M(P))$, therefore in $M(P')$ there is an outgoing edge from $A$ to all the other candidates and thus $A$ is a Condorcet winner for $P'$, and therefore a possible Condorcet winner for $P$.

Now for point 4, the fact that any necessary Condorcet winner from a majority graph $M(P)$ is a necessary Condorcet winner for the profile $P$ is an obvious consequence of Proposition 1. For the converse inclusion, let suppose $A$ is not a necessary Condorcet winner for $M(P)$. Then there exists $B$ such that $(A, B) \notin M(P)$, that is, $\#\{i \mid A >_i B\} < \frac{n}{2}$. Let us complete the profile by adding $B >_j A$ for all voters $j$ for which the relation between $A$ and $B$ is missing: we get an extension of $P$ in which a majority of voters prefers $B$ to $A$, therefore $A$ is not a necessary Condorcet winner for $P$. Hence, in the (possibly incomplete) majority graph $M(P)$ induced by $P$, there are outgoing edges from this candidate to every other candidate, and so this candidate must belong to the necessary Condorcet winners for $M(P)$. □

Notice that there are cases in which the subset relation $Poss(M(P), r_{Schwartz}) \supseteq Poss(P, r_{Schwartz})$ is strict. In fact, a candidate can be a Schwartz winner for a completion of $M(P)$ which is not induced by any completion of $P$, as shown previously in Example 8.

Note that if $r$ is a Condorcet-consistent rule, then $NecCond(P) \subseteq Nec(P, r)$ and $PossCond(P) \subseteq Poss(P, r)$.

#### 3.2.2 Weighted profiles.

Remark that Proposition 2 extends easily to weighted profile, that is, $Poss(M(P), r) \supseteq Poss(P, r)$, $Nec(M(P), r) \subseteq Nec(P, r)$, $NecCond(M(P)) = NecCond(P)$, and $PossCond(M(P)) = PossCond(P)$. In fact, obviously, $M(P) = M(U(P))$ and it is easy to show that $NecCond(P) = NecCond(U(P))$ and $PossCond(P) = PossCond(U(P))$.

# 4 Computing winners with incomplete preferences in SMV(T)

We now analyze the complexity of computing the various kinds of winners in the sequential majority voting rule $SMV(T)$ when preferences are incomplete.

We will consider first winners from an incomplete profile and then winners from an incomplete majority graph.

## 4.1 Incomplete profiles

Assume agents' preferences have been specified by an incomplete weighted profile $P$. It is possible to show that, when we have 3 or more candidates, it is difficult to determine if a candidate is a possible winner.

**Theorem 2** *Given a fixed number $n$ of candidates with $n \geq 3$, an incomplete weighted profile $P$ and any agenda $T$ on these candidates, deciding if a candidate is in $Poss(P, r_{SMV(T)})$ is NP-complete.*

**Proof:** We give a reduction from the number partitioning problem. Consider the tree $T$ where $A$ plays against $B$ and the winner thus plays against $C$. We have a bag of integers, $k_i$ with sum $2k$ and we wish to decide if they can be partitioned into two bags, each with sum $k$. We will show that we can build an election where we can complete the incomplete weighted profile so that $C$ wins (i.e., $C$ is a possible winner) iff such a partition exists. We suppose the following votes are given: 1 vote for $C > B > A$ of weight 1, 1 vote $C > A > B$ of weight $2k - 1$, and 1 vote $B > C > A$ of weight $2k - 1$. Hence, $C$ is ahead of $A$ by $4k - 1$ votes, $C$ is ahead of $B$ by 1 vote and $B$ is ahead of $A$ by 1 vote. For each $k_i$ in the bag of integers, we have an incomplete vote of weight $2k_i$ in which $A > C$ is fixed, but the rest of the vote is incomplete. We are sure $A$ beats $C$ in the final result by 1 vote whatever completion takes place. We now show that the incomplete weighted profile can be completed so that $B$ beats $A$ and $C$ beats $B$ (an thus $C$ is a possible winner) iff there is a partition of size $k$.

Assume that there is such a partition exists, and that votes in one partition have $A > C > B$ and the votes in the other have $B > A > C$. Thus, $B$ beats $A$ overall and $C$ beats $B$. Thus $C$ is the winner.

On the other hand, suppose there is a way to complete the preferences so that $C$ wins. This can only happen if $B$ beats $A$ and $C$ then beats $B$. In fact, if $A$ beats $B$ in the first round, $A$ will beat $C$ in the second round, since we have shown before that $A$ beats $C$ overall, and then $A$ will be the final winner. For $C$ to beat $B$, at least half the weight of incomplete votes must rank $C$ above $B$. Similarly, for $B$ to beat $A$, at least half the weight of incomplete votes must rank $B$ above $A$. Since all votes rank $A$ above $C$, $B$ cannot be both above $A$ and below $C$. Thus precisely half the weight of incomplete votes ranks $B$ above $A$ and half ranks $C$ above $B$. Hence, we have a partition of equal weight. Therefore, we can complete the incomplete profile so that $C$ wins iff there is a partition of size $k$. The result can also be extended to the trees with more than 3 candidates, since they extend $T$ or a tree isomorphic to it, by placing any additional candidate at the bottom of every voters' preference ordering (it does not matter how). $\square$

We conjecture that determining whether a candidate is a possible/necessary winner for SMV(T) is difficult also when we have unweighted votes.

We now show that, when we have 4 or more candidates, it is a coNP-complete problem to determine if a candidate is a necessary winner.

**Theorem 3** *Given a fixed number $n$ of candidates with $n \geq 4$, an incomplete weighted profile $P$ and any agenda $T$ on these candidates, deciding if a candidate is in $Nec(P, r_{SMV(T)})$ is coNP-complete.*

**Proof:** To show that with more than 4 candidates it is coNP-complete to decide if a candidate wins for every completion of the preferences, we will consider the tree $T$ where $A$ plays against $B$, the winner then plays against $C$, and the winner of this match goes forward to the final match against $D$. We will reduce number partitioning to deciding if, given a particular incomplete weighted profile, we can complete such a profile to make a candidate a necessary winner.

We have a bag of integers, $k_i$ with sum $2k$ and we wish to decide if they can be partitioned into two bags, each with sum $k$. We construct an incomplete profile where the following votes are given: 1 vote for $C > D > B > A$ of weight 1, 1 vote $C > D > A > B$ of weight $2k - 1$, and 1 vote $D > B > C > A$ of weight $2k - 1$. For the first number, $k_1$ in the bag of integers, we have a vote for $D > B > A > C$ of weight $2k_1$. For each other number, $k_i$ where $i > 1$, we have an incomplete vote of weight $2k_i$ in which $A > C$ is fixed, but the rest of the vote is incomplete. We are sure that $A$ beats $C$ in the final result by 1 vote whatever completion takes place. Similarly, we are also sure that $D$ beats $A$, and $D$ beats $B$.

If we complete preferences so that in all incomplete votes we have $D > C$, then $D$ will win overall. We now show that there is a completion that makes $C$ win iff there is a partition of equal weight.

Suppose there is such a partition and that the complete votes in one partition have $B > A > C$ and the complete votes in the other have $A > C > B$. Thus, $B$ beats $A$ overall, $C$ beats $B$. We suppose also that enough of the incomplete votes are completed with $C > D$ for $C$ to beat $D$. Thus $C$ is the overall winner.

On the other hand, suppose there is a way to complete the preferences so that $C$ wins. This can only happen if $B$ beats $A$, $C$ then beats $B$ and $C$ finally beats $D$. If $A$ beats $B$ in the first round, $A$ will beat $C$ in the second round and then go out to $D$. For $C$ to beat $B$, at least half the weight of incomplete votes must rank $C$ above $B$. Similarly, for $B$ to beat $A$, at least half the weight of incomplete votes must rank $B$ above $A$. Since all votes rank $A$ above $C$, $B$ cannot be both above $A$ and below $C$. Thus precisely half the weight of incomplete votes ranks $B$ above $A$ and half ranks $C$ above $B$, and thus we have a partition of equal weight. Hence, if there is a partition of equal weight then both $C$ and $D$ are possible winners; if not, then only $D$ is a possible winner, therefore a necessary winner. Therefore, $D$ is a necessary winner if and only if there is no partition in the initial problem. The proof can be extended to all the trees with more than 4 candidates, since they extend $T$ or a tree isomorphic to it, by placing any additional candidate at the bottom of every voter's preference ordering (it does not matter how). $\square$

These results can be compared with Theorem 7 in [3] which states that weighted constructive manipulation for $SMV(T)$ is polynomial. A candidate can be made a winner by a coalition of weighted votes if it is a possible winner for a specific incomplete profile consisting of full votes and empty votes. By allowing any kind of incomplete profiles, we get a complexity shift.

The possible winner problem and the necessary winner problem for SMV($T$), when $T$ is balanced, have been also considered in [20]. In [20] it is shown that, in the case of unweighted profiles and an unbounded number of candidates, the possible winner problem is NP-complete, while the necessary winner problem is coNP-complete. We have shown the same results but in a different context, i.e., in the case of weighted profiles and a fixed number of candidates.

## 4.2 Incomplete majority graphs

We now present an algorithm, called *Win*, for determining, given an agenda $T$ and an incomplete majority graph $G$, the set of possible winners (i.e., $Poss(G, r_{SMV(T)})$). We will represent an agenda $T$ with a binary a tree with a root, that we call $root(T)$, a left subtree, called $left(T)$, and a right subtree, called $right(T)$.

Algorithm *Win* (see the pseudocode below) recursively takes in input an agenda $T$, an incomplete majority graph $G$, and it returns a set of candidates $W$, which is the set of possible winners If $root(T)$ is not empty, and both $left(T)$ and $right(T)$ are empty, then the algorithm returns $root(T)$, otherwise, the set $W_1$ (resp., $W_2$) is initialized with the set of candidates that can win in some completion of $G$ in the left (resp., right) subtree of $T$. Next, set $W$ is initialized with the candidates of $W_1 \cup W_2$. Then, the algorithm removes from $W$ every $s \in W_1$ (resp., $r \in W_2$) that is worse than all the other elements $r$ of $W_2$ (resp., $s$ of $W_1$) in $G$, i.e., the candidates $s \in W_1$ (resp. $r \in W_2$) such that $s <_m r$, $\forall r \in W_2$ (resp., such that $r <_m s$, $\forall s \in W_1$). Finally, it returns the set $W$, that will contain the set of possible winners.

---

**Algorithm 1**: *Win*

---

```
Input: T: an agenda, G: an incomplete majority graph;
Output: W: set of candidates;
if root(T) ≠ nil and left(T) = right(T) = nil then
    W ← label(root(T));
else
    W1 ← Win(left(T), G);
    W2 ← Win(right(T), G);
    W = W1 ∪ W2;
    foreach s ∈ W1 do
        if s <m r, ∀r ∈ W2 then
            W ← W \ {s}
    foreach r ∈ W2 do
        if r <m s, ∀s ∈ W1 then
            W ← W \ {r}
return W;
```

---

To check whether there exists a necessary winner for an incomplete majority graph $G$, it suffices to run the algorithm $Win$ on $P$; if it outputs a single candidate, then it is a necessary winner, otherwise there is no necessary winner. We will call *NecessaryWin* such a procedure.

**Example 10** *We now show how to determine possible and necessary winners given a fixed agenda a applying* Win.

*Consider the set of candidates* $\Omega = \{A, B, C, D, E, F, H, I\}$. *Consider the agenda* $T$ *over* $\Omega$ *with* $left(root(T)) = Win(Win(\{C\}, \{D\}), Win(\{E\}, \{F\}))$ *and* $right(root(T)) = Win(Win(\{A\}, \{B\}), Win(\{I\}, \{H\}))$. *Consider also the* *incomplete majority graph $G$ with edges $A >_m B$, $A >_m C$, $A >_m D$, $A >_m E$, $A >_m I$, $E >_m F$, and $I >_m H$.*

*Algorithm* Win *works as follows. First, it computes the possible winners in the left subtree of $T$. To do that, it computes $Win(\{C\}, \{D\})$, and it returns the set $\{C, D\}$, since there is no edge in the majority graph between $C$ and $D$, that means that both $C$ and $D$ win in at least a completion. In fact, if we consider the completion of the majority graph where $C >_m D$, then $C$ wins, while, if we consider the completion of the majority graph where $D >_m C$, then $D$ wins. Similarly, it computes $Win(\{E\}, \{F\})$. In this case it returns the set $\{E\}$, since in the majority graph there is an edge $E >_m F$. Next, it computes $Win(Win(\{C\}, \{D\}), Win(\{E\}, \{F\}))$, i.e., $Win(\{C, D\}, \{E\})$, and it returns the set $\{C, D, E\}$, since $E$ is not beaten by both $C$ and $D$, $C$ is not beaten by $E$ and $D$ is not beaten by $E$.*

*Next, it computes the possible winners in the right subtree of $T$. To do that, it computes $Win(\{A\}, \{B\})$, and it returns the set $\{A\}$, since we have $A >_m B$ in the majority graph. Next, it computes $Win(\{I\}, \{H\})$ and it returns the set $\{I\}$, since there is $I >_m H$. After that, it computes $Win(Win(\{A\}, \{B\}), Win(\{I\}, \{H\}))$, i.e., $Win(\{A\}, \{I\})$, and it returns the set $\{A\}$, since in the majority graph we have $A >_m I$.*

*Then, it performs the union between the set of possible winners of the left subtree (i.e., $\{C, D, E\}$) and those of the right subtree (i.e., $\{A\}$), and it removes from such union all the possible winners of the left subtree, since they are all beaten by $A$ in the majority graph. This means that in every completion of the majority graph $C$, $D$, and $E$ cannot win. Thus,* Win *returns $\{A\}$. Since the set retuned by* Win *contains only one candidate, such a candidate is also the necessary winner.* □

**Theorem 4** *Consider a set of candidates $\Omega$, an agenda $T$ over $\Omega$, and a possibly incomplete majority graph $G$ over $\Omega$. Then,*

- *Algorithm* Win$(T, G)$ *returns $Poss(G, r_{SMV(T)})$.*
- *Algorithm* NecessaryWin$(T, G)$ *returns $Nec(G, r_{SMV(T)})$.*
- *Both algorithms run in polynomial time in the number of candidates.*

**Proof:** Let us first consider Algorithm *Win*. We will prove its soundness and completeness by induction on the structure of the tree.

For the soundness we need to prove that, for every candidate $A$, if $A \notin Poss(G, r_{SMV(T)})$, then $A$ is not returned by $Win(T, G)$. If the tree has one level, that is, only one node, it means there is only one candidate. The node of the tree contains the only candidate, which is returned by the algorithm and it is also a possible winner. If the tree has two levels, that is, there are three nodes (we recall that the tree is binary), it means that there are two candidates, $A$ and $B$. Let us assume that $A \notin Poss(G, r_{SMV(T)})$. This means that $G$ contains (only) the edge $B >_m A$. The tree has three nodes: a root and two leaves containing the two candidates. If $B >_m A$, Algorithm *Win* will bring to the root level, and thus return, only $B$. Thus $A$ is not retuned by *Win*. Assume now to consider a tree $T$ with $n$ levels. By inductive hypothesis, we assume the statement holds for trees with $n-1$ levels. Thus, for both $right(T)$ and $left(T)$, we know that, if

$A \notin Poss(G, r_{SMV(left(T))})$, then $A$ is not returned by $Win(left(T), G)$. The same holds for $right(T)$. Assume $A \notin Poss(G, r_{SMV(T)})$. There are two cases: either $A \in (Win(left(T), G) \cup Win(right(T), G))$, or not. In the second case $Win(T, G)$ does not return $A$. In the first case, assume $A \in Win(left(T), G)$ and $A \in Win(T, G)$. According to algorithm *Win*, there exists a candidate $B \in Win(right(T), G)$ that does not beat $A$. This means that the majority graph does not contain $B >_m A$. By inductive hypothesis, there is a completion of the majority graph such that $B$ wins in $right(T)$. This completion can be further extended by adding, if necessary, $A >_m B$ in the majority graph. Thus, in this completion $A$ wins in $T$. Thus $A \in Poss(G, r_{SMV(T)})$. A similar inductive reasoning can be followed to prove that Algorithm *Win* is complete, that is, that $A \in Poss(G, r_{SMV(T)})$ implies $A \in Win(T, G)$.

Algorithm *NecessaryWin* is trivially correct and complete since a candidate is a necessary winner iff it is the only possible winner.

If we have $n$ candidates, Algorithm *Win* performs at most $O(n)$ steps, where at each step it may consider $O(n^2)$ pairs of candidates. For each pair it performs a constant work, that is, checking the presence of one edge in the majority graph. Thus, the overall complexity is $O(n^3)$. The same complexity holds for Algorithm *NecessaryWin*, since it just calls Algorithm *Win* and then adds one more constant time step. □

Since $Nec(M(P), r_{SMV(T)}) \subseteq Nec(P, r_{SMV(T)})$ and $Poss(M(P), r_{SMV(T)}) \supseteq Poss(P, r_{SMV(T)})$, the set returned by Algorithm *Win* is an upper approximation of $Poss(P, r_{SMV(T)})$.

# 5 Computing Schwartz winners with incomplete preferences

We analyze the complexity of determining Schwartz winners, when we have uncertainty in the preferences.

We now show that computing possible Schwartz winners from incomplete weighted profiles is intractable in general, while it is polynomial if we consider the induced majority graphs. Hence, computing the winners for incomplete majority graphs can be a cheap way to have an approximation of the winners for incomplete weighted profiles. We conjecture that determining possible Schwartz winners for incomplete profiles is NP-hard as well, but so far we don't have a proof.

## 5.1 Incomplete weighted profiles

We now show that computing possible Schwartz winners for weighted profiles is NP-hard for incomplete weighted profiles where there are 3 or more candidates.

**Theorem 5** *Given a fixed number $n$ of candidates with $n \geq 3$, an incomplete weighted profile $P$ on these candidates, deciding if a candidate is in $Poss(P, r_{Schwartz})$ is NP-complete.*

**Proof:** Clearly the problem is in NP as a polynomial witness is a completion and an agenda in which the candidate wins. To show it is NP-complete, we give a reduction from the number partitioning problem. The reduction is based around constructing a Condorcet cycle and is similar to those used in [3] to show that manipulation is computationally hard even with a small number of candidates when votes are weighted.

We have a bag of integers, $k_i$ with sum $2k$ and we wish to decide if they can be partitioned into two bags, each with sum $k$. We want to show that a candidate $B$ is a possible Schwartz winner if and only if such a partition exists. We construct an incomplete profile over three candidates ($A$, $B$, and $C$) as follows. We have 1 vote for $B > C > A$ of weight 1, 1 vote $B > A > C$ of weight $2k-1$, and 1 vote $C > B > A$ of weight $2k-1$. At this point, the total weight of votes with $B > A$ exceeds that of $A > B$ by $4k-1$, the total weight of votes with $B > C$ exceeds that of $C > B$ by 1, and the total weight of votes with $C > A$ exceeds that of $A > C$ by 1.

We also have, for each $k_i$, a partially specified vote of weight $2k_i$ in which we know just that $A > B$. As the total weight of these partially specified votes is $4k$, we are sure $A$ beats $B$ in the final result by 1 vote. The only agenda in which $B$ can win is the one in which $A$ plays $C$ and the winner then plays $B$. In addition, for $B$ to win, the partially specified votes need to be completed so that $B$ beats $C$, and $C$ beats $A$ in the final result. We show that this is possible iff there is a partition of size $k$. Suppose there is such a partition. Then let the votes in one bag be $A > B > C$ and the votes in the other be $C > A > B$. Then, $A$ beats $B$, $B$ beats $C$ and $C$ beats $A$, all by 1 vote in the final result. On the other hand, suppose there is a way to cast the votes to give the result $A$ beats $B$, $B$ beats $C$ and $C$ beats $A$. All the uncast votes rank $A$ above $B$. In addition, at least half the weight of votes must rank $B$ above $C$, and at least half the weight of votes must rank $C$ above $A$. Since $A$ is above $B$, $C$ cannot be both above $A$ and below $B$. Thus precisely half the weight of votes ranks $C$ above $A$ and half ranks $B$ above $C$. Hence we have a partition of equal weight. We therefore conclude that $B$ can win iff there is a partition of size $k$. That is, deciding if $B$ is a possible Schwartz winner is NP-complete. We can extend the reduction to more than 3 candidates by placing any additional candidate at the bottom of every voters' preference ordering (it does not matter how). □

Computing possible Schwartz winners for weighted profiles is difficult, however we will show that is easy to find an upper approximations of these winners considering the majority graph.

## 5.2 Incomplete majority graphs

We now give graph characterizations of possible and necessary Schwartz winners for incomplete majority graphs, that allow us to conclude that they are polynomial to compute.

**Theorem 6** *Given an incomplete majority graph $G$ and a candidate $A$, $A \in Nec(G, r_{Schwartz})$ if and only if, for every other candidate $B$, there is a path from $A$ to $B$ in $G$.*

**Proof:** ($\Leftarrow$) Suppose that for each $B \neq A$ there is a path from $A$ to $B$ in $G$. Then these paths remain in every completion of $G$. Therefore, using Theorem 1, $A$ is

a Schwartz winner in every completion of $G$, i.e., it is a necessary Schwartz winner.
($\Rightarrow$) Suppose there is no path from $A$ to $B$ in $G$. Let us define the following three subsets of the set of candidates $\Omega$: $R(A)$ is the set of candidates reachable from $A$ in $G$ (including $A$); $R^{-1}(B)$ is the the set of candidates from which $B$ is reachable in $G$ (including $B$); and $Others = \Omega \setminus (R(A) \cup R^{-1}(B))$. Because there is no path from $A$ to $B$ in $G$, we have that $R(A) \cap R^{-1}(B) = \emptyset$ and therefore $\{R(A), R^{-1}(B), Others\}$ is a partition of $\Omega$. Now, let us build the complete tournament $\hat{G}$ as follows:

1. $\hat{G} := G$;
2. $\forall\, x \in R(A), y \in R^{-1}(B)$, add $(y, x)$ to $\hat{G}$;
3. $\forall\, x \in R(A), y \in Others$, add $(y, x)$ to $\hat{G}$;
4. $\forall\, x \in Others, y \in R^{-1}(B)$, add $(y, x)$ to $\hat{G}$;
5. $\forall\, x, y$ belonging to the same element of the partition: if neither $(x, y)$ nor $(y, x)$ is in $G$ then add one of them (arbitrarily) in $\hat{G}$.

Let us first show that $\hat{G}$ is a complete tournament. If $x \in R(A)$ and $y \in R^{-1}(B)$, then $(x, y) \notin G$ (otherwise there would be a path from $A$ to $B$ in $G$). If $x \in R(A)$ and $y \in Others$, then $(x, y) \notin G$, otherwise $y$ would be in $R(A)$. If $x \in Others$ and $y \in R^{-1}(B)$, then $(x, y) \notin G$, otherwise $x$ would be in $R^{-1}(B)$. Therefore, whenever $x$ and $y$ belong to two distinct elements of the partition, $\hat{G}$ contains $(y, x)$ and not $(x, y)$. Now, if $x$ and $y$ belong to the same element of the partition, by Step 5, $\hat{G}$ contains exactly one edge among $\{(x, y), (y, x)\}$. Therefore, $\hat{G}$ is a complete tournament. Let us show now that there is no path from $A$ to $B$ in $\hat{G}$. Suppose there is one, that is, there exist $z_0 = A, z_1, \ldots, z_{m-1}, z_m = B$ such that $\{(z_0, z_1), (z_1, z_2), \ldots, (z_{m-1}, z_m)\} \subseteq \hat{G}$. Now, for all $x \in R(A)$ and all $y$ such that $(x, y) \in \hat{G}$, by construction of $\hat{G}$ we necessarily have $y \in R(A)$. Therefore, for all $i < m$, if $z_i \in R(A)$ then $z_{i+1} \in R(A)$. Now, since $z_0 = A \in R(A)$, by induction we have $z_i \in R(A)$ for all $i$, thus $B \in R(A)$, which is impossible. Therefore, there is no path from $A$ to $B$ in $\hat{G}$. Thus, $\hat{G}$ is a complete tournament with no path from $A$ to $B$, which implies that $A$ is not a Schwartz winner w.r.t. $\hat{G}$. Lastly, by construction, $\hat{G}$ contains $G$. So $\hat{G}$ is a complete extension of $G$ for which $A$ is not a Schwartz winner. This shows that $A$ is not a necessary Schwartz winner for $G$. □

Clearly, a procedure based on the previous theorem gives us a polynomial algorithm to find necessary Schwartz winners. Hence, we have a polynomial approximation of necessary Schwartz winners for incomplete profiles.

**Corollary 1** *Given an incomplete majority graph $G$, $Nec(G, r_{Schwartz})$ can be computed in polynomial time.*

We will now pass to consider possible Schwartz winners. Let $G$ an asymmetric graph, $\Omega$ the set of candidates, and $A \in \Omega$. Let us consider Algorithm 2. It tries to complete $G$ in order to have a path from $A$ to every other candidate $Z$ by putting edges directed to $Z$, where it is possible, from the candidates that are reachable from $A$.

**Algorithm 2**: PossibleSchwartzWinners

**Input**: $G$: an asymmetric graph, $A$: a candidate,
**Output**: $\Sigma$: set of candidates;
$\Sigma \leftarrow \{A\} \cup \{X|$ there is a path from $A$ to $X$ in $G\}$;
$G' \leftarrow G$;
**repeat**
  **foreach** $(Y, Z) \in \Sigma \times (\Omega \setminus \Sigma)$ **do**
    **if** $(Z \to Y) \notin G'$ **then**
      add $(Y \to Z)$ to $G'$;
  **foreach** $Z \in \Omega \setminus \Sigma$ **do**
    **if** *there is a path from $A$ to $Z$ in $G'$* **then**
      add $Z$ to $\Sigma$;
**until** $\Sigma = \Omega$ *or there is no* $(Y, Z) \in \Sigma \times (\Omega \setminus \Sigma)$ *s. t.* $(Z \to Y) \in G'$ ;
**return** $\Sigma$

**Theorem 7** *Given an incomplete majority graph G and a candidate A,* $A \in Poss(G, r_{Schwartz})$ *if and only if* PossibleSchwartzWinners$(G, A) = \Omega$.

**Proof:** We first make the following observation: the graph $G'$ obtained at the end of the algorithm is asymmetric and extends $G$. It is asymmetric because it is so at the start of the algorithm (since $G$ is) and then, when an edge $Y \to Z$ is added to $G'$ when $Z \to Y$ is not already in $G'$.

Now, assume PossibleSchwartzWinners$(G, A) = \Omega$. Let $G''$ be a tournament extending $G'$ (and, a fortiori, $G$). Such a $G''$ exists (because $G'$ is asymmetric). By construction of $G'$, there is a path in $G'$ from $A$ to every node of $f(G, A) \setminus \{A\}$, hence to every node of $\Omega \setminus \{A\}$; since $G''$ extends $G'$, this holds a fortiori for $G''$, hence $A$ is a Schwartz winner in $G''$ and therefore a possible Schwartz winner for $G$.

Conversely, assume PossibleSchwartzWinners$(G, A) = \Sigma \neq \Omega$. Denote $\Theta = \Omega \setminus \Sigma$. Then, for all $(Y, Z) \in \Sigma \times \Theta$ we have $Z \to Y \in G'$. Now, $Z \in \Theta$ means that no edge $Z \to Y$ (for $Z \in \Theta$ and $Y \in \Sigma$) was added to $G'$; hence, for every $Y \in \Sigma$ and $Z \in \Theta$, we have that $Z \to Y \in G'$ if and only if $Z \to Y \in G$. This implies that for all $(Y, Z) \in \Sigma \times \Theta$ we have $Z \to Y \in G$, therefore, in every tournament $G''$ extending $G$, every candidate of $\Theta$ beats every candidate of $\Sigma$, and in particular $A$. Therefore, there cannot be a path in $G''$ from $A$ to a candidate in $Z$, which implies that $A$ is not a Schwartz winner in $G''$. Since the latter holds for every tournament $G''$ extending $G$, $A$ is not a possible Schwartz winner for $G$. □

As a consequence of Theorem 7, possible Schwartz winners are computable in polynomial time, which means that they can be seen as a polynomial approximation of possible Schwartz winners for incomplete profiles.

**Corollary 2** *Given an incomplete majority graph G, the set* $Poss(G, r_{Schwartz})$ *can be computed in polynomial time.*

Given an asymmetric graph $G$, $\Theta$ is said to be a *dominant subset* of $G$ if and only if, for every $Z \in \Theta$ and every $X \in \Omega \setminus \Theta$, we have $(Z, X) \in G$. Then we have an alternative characterization of possible Schwartz winners:

**Theorem 8** *Given an incomplete majority graph G and a candidate A, $A \in Poss(G, r_{Schwartz})$ if and only if A belongs to all dominant subsets of G.*

**Proof:** Suppose there exists a dominant subset $\Theta$ of $G$ such that $A \notin \Theta$. Then there can be no extension of $G$ in which there is a path from $A$ to a candidate $Z \in \Theta$. Hence $A$ is not a possible Schwartz winner for $G$.

Conversely, suppose that $A$ is not a possible Schwartz winner for $G$. Then the algorithm for computing PossibleSchwartzWinners$(G, A)$ stops with PossibleSchwartzWinners $(G, A) \neq \Omega$ and $\Omega\setminus$ PossibleSchwartzWinners$(G, A)$ being a dominant subset of $G$. Since $A \in$ PossibleSchwartzWinners$(G, A)$, $\Omega\setminus$ PossibleSchwartzWinners$(G, A)$ is a dominant subset of $G$ to which $A$ does not belong. $\square$

# 6 Computing winners over balanced agendas

We now turn our attention to fair winners, that is, candidates that win in some balanced agenda. Here, we start by considering the computation of FSMV with complete preferences, as this is a nontrivial problem.

## 6.1 Computing fair winners

For simplicity, we will assume there are $2^k$ candidates. We will show that a candidate is a fair winner iff the nodes of the majority graph can be covered by a binomial tree [6].

Binomial trees are defined inductively as follows:

- A binomial tree of order 0, written $T_0$, is a tree with only one node.
- A binomial tree of order $k$, with $k > 0$, written $T_k$, is the tree where the root has $k$ children, and for $i = 1, \ldots, k$ the i-th child is a binomial tree of order $k - i$.

It is easy to see that, in a binomial tree of order $k$, there are $2^k$ nodes and the tree has height $k$.

Given a majority graph with $2^k$ nodes, and given a candidate $A$, it may be possible to find a covering of the nodes which is a binomial tree of order $k$ with root $A$. In this situation, we can build a balanced binary tree where $A$ wins. Thus $A$ is a fair winner.

**Theorem 9** *Given a majority graph G with $2^k$ nodes, and a candidate A, there is a binomial tree $T_k$ covering all nodes of G with root A, if and only if A is a fair winner.*

**Proof:** Assume there is a binomial tree $T_k$ satisfying the statement of the theorem. Then each node of $T_k$ is associated to a candidate. For node $n$, we will write $cand(n)$ to denote such a candidate.

We will show that it is possible to define, starting from $T_k$, a balanced binary tree $b(T_k)$ where $A$ wins. Each node of $b(T_k)$ is associated to a candidate as well, with the same notation as above. The definition of $b(T_k)$ is given by induction:

- If $k = 0$, $b(T_0) = T_0$;
- If $k > 0$, $b(T_k)$ is the balanced tree built from two instances of $b(T_{k-1})$, corresponding to the two instances of $T_{k-1}$ which are part of $T_k$ by definition; the roots of such trees are the children of the root of $b(T_k)$; the candidate of the root of $b(T_k)$ is the candidate of the root of $T_k$.

It is easy to see that $b(T_k)$ is a perfectly balanced binary tree, where the winner is $A$ if $A$ is the candidate associated to the root of the binomial tree. Proof of the opposite direction is similar. $\square$

**Example 11** *Consider the majority graph $G$ over candidates $A$, $B$, $C$ and $D$ depicted in Figure 2. Since such a majority graph is covered by the binomial tree $T_2$ with root $A$, we can conclude that $A$ is a fair winner.* $\square$

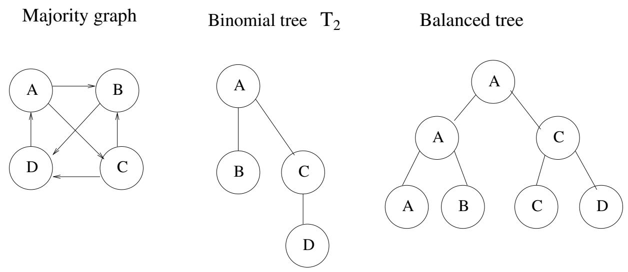


Figure 2: From a majority graph to a balanced binary tree via a binomial tree.

The complexity of deciding if $A$ is a fair winner is an open problem. However, for *weighted* majority graphs, it is NP-hard to check whether a candidate $A$ is a fair winner. This means that, if we restrict the agendas only to balanced ones, it is difficult for the chair to control the election.

**Theorem 10** *Given a complete weighted majority graph $G$ and a candidate $A$, it is $\mathcal{NP}$-complete to check whether $A$ is a fair winner for $G$.*

**Proof:** We prove that the Exact Cover problem reduces polynomially to the problem of finding a minimum binomial tree with root $A$ covering $G$. The proof is similar to the one in [13] that reduces Exact Cover to the problem of finding a minimum spanning tree for a graph from a class of trees which are sets of disjoint flowers of type 2 and where, for each tree $t$ in the class, the number of flowers of type 2 is $d \geq c|t|^{\epsilon}$.

A flower is a tree where all nodes but the root have degree at most 2 and are at distance 1 or 2 from the root. A flower is of type 2 if at least a node has distance 2 from the root.

Binomial trees are trees which consist of disjoint flowers of type 2. In fact, the binomial tree $T_2$ is a flower of type 2, and all bigger binomial trees $T_k$, with $2^k$ nodes, consist of $2^{k-2}$ instances of such flowers of type 2, all disjoint[3]. Thus we have $d = 2^{k-2} = \frac{|T_k|}{2} \geq c|T_k|^\epsilon$.

Given an instance of the Exact Cover Problem, consisting of $3k$ sets of size 3 and $3k$ elements, each of which appears three times in the sets, the proof proceeds by constructing a graph such that the instance has an exact cover iff the graph has a minimum covering binomial tree. □

On the other hand, if $A$ is a Schwartz winner, it is easy to find a tree with a *bounded level of imbalance* in which $A$ wins. We define $D(T, A)$ as the length of the path in the tree $T$ from the root to the only leaf labelled $A$. We define $\Delta(A)$ as the maximum $D(T, A)$ over all competition trees $T$ where $A$ wins. Obviously, $\Delta(A) = m - 1$ *iff* $A$ is a Condorcet winner.Computing $\Delta(A)$ is easy:

**Proposition 3** *Given a majority graph G and a Schwartz winner A, $\Delta(A)$ is the number of outgoing edges from A in G.*

**Proof:** If $A$ has $k$ outgoing edges, no binary tree where $A$ wins can have $A$ appearing at depth larger than $k$. In fact, to win, $A$ must win in all competitions scheduled by the tree, so such competitions must be at most $k$. Thus $\Delta(A) \leq k$.

Moreover, it is possible to build a binary tree where $A$ wins and appears at depth $k$. Let us first consider the linear tree, $T_1$ in which $A$ competes against all and only the $D_1, \ldots, D_k$ candidates which it defeats directly in $G$. Clearly in such tree, $T_1$, $D(T_1, A) = k$. However $T_1$ may not contain all candidates. In particular it will not contain candidates defeating $A$ in $G$. We now consider, one by one, each candidate $C$ such that $C \to A$ is an edge of $G$. For each such candidate we add a subtree to the current tree. The current tree at the beginning is $T_1$. Let us consider the first $C$ and the path, which we know exists, which connects $A$ to $C$, say $A \to B_1 \to B_2 \ldots \to B_h \to C$. Let $j \in \{1, \ldots, h\}$ be such that $B_j$ belongs to the current tree and $\forall i > j$, $B_i$ does not belong to the current tree. Notice that such candidate $B_j$ always exists, since any path from $A$ must start with an edge to one of the $D_1, \ldots, D_k$ candidates. We then attach to the current tree the subtree corresponding to the path $B_j \to \ldots \to C$ at node $B_j$ obtaining a new tree in which only new candidates have been added. After having considered all candidates defeating $C$ in $A$, the tree obtained is a binary tree in which $A$ wins and has depth exactly $k$. Thus $\Delta(A) = k$. □

Note that if $A$ is not a Schwartz winner, this equality does not necessarily hold.

We now show that, if $A$ is a Schwartz winner, there exists a binary tree where $A$ wins with level of imbalance at most $m - \Delta(A) - 1$. Notice that there could also be more balanced trees in which $A$ wins.

**Proposition 4** *Given a majority graph G and a Schwartz winner A, there is a binary tree with level of imbalance smaller than or equal to $m - \Delta(A) - 1$, where A wins. This tree can be built in polynomial time.*

[3]We thank Claude-Guy Quimper for this observation and for suggesting us to look at [13].

**Proof:** Since $A$ beats $\Delta(A)$ candidates, we can easily build a balanced binary tree $BT$ involving only $A$ and the candidates beaten by $A$. Then we have to add the remaining $m - \Delta(A) - 1$ candidates to $BT$. Since $A$ is a Schwartz winner, there must be a path from one of the candidates beaten by $A$ to each of the remaining candidates. In the worst case, this adds a subtree of depth $m - \Delta(A) - 1$ rooted at one of the nodes beaten by $A$ in $BT$. $\square$

If unfair tournaments are undesirable, we can consider those Schwartz winners for which there are binary trees which are as balanced as possible. Proposition 4 helps us in this respect: if $A$ is a Schwartz winner, knowing $\Delta(A)$ we can compute an upper bound to the minimum imbalance of a tree where $A$ wins. In general, if $\Delta(A) \geq k$, then there is a tree with imbalance level smaller or equal than $m - k - 1$, hence the higher $\Delta(A)$ is, the lower is the upper bound to the level of imbalance of a tree where $A$ wins.

If a candidate has the maximum number of outgoing edges in the majority graph (i.e., it is a *Copeland winner* [12, 11]), we can give a smaller upper bound on the amount of imbalance in the fairest/most balanced tree in which it wins. Notice that every Copeland winner is a Schwartz winner.

**Proposition 5** *If a candidate $A$ is a Copeland winner, then the imbalance of a fairest/most balanced tree in which $A$ wins is smaller or equal than $log(m - \Delta(A) - 1)$.*

**Proof:** If $A$ is Copeland winner then every candidate beating $A$ must be beaten by at least one candidate beaten by $A$. Consider the balanced tree $BT$ defined in the proof of Proposition 4 rooted at $A$ and involving all and only the candidates beaten by $A$. In the worst case all the remaining $m - \Delta(A) - 1$ candidates are beaten only by the same candidate in $BT$, say $B'$. In such case, however, we can add to $BT$ a balanced subtree with depth $log(m - \Delta(A) - 1)$ rooted at $B'$, involving all the remaining candidates. $\square$

## 6.2 Computing fair winners with incomplete preferences

We now show that determining the winner in SMV(T) is difficult when we have weighted votes also when we consider a balanced tree. We conjecture that it continues to be difficult also when we have unweighted votes.

**Theorem 11** *Given a fixed number $n$ of candidates with $n \geq 3$, an incomplete weighted profile $P$ and a balanced agenda $T$ on these candidates, determining the winner in SMV($T$) is NP-complete.*

**Proof:** We can use the same construction in the proof of Theorem 5. Given the profile constructed there, the only possible balanced tree in which $C$ wins are those in which $A$ plays $B$, and (at some later point) $C$ then plays the winner. All the additional candidates will be defeated by $A$, $B$ and $C$ so they can be placed anywhere in the balanced tree. $\square$

We recall that the candidates that win in every balanced agenda coincide with Condorcet winners, thus the candidates that win in every balanced agenda for some completion (resp., all completions) of the preferences coincide with possible (resp., necessary) Condorcet winners, and thus they are polynomial to find both for weighted and unweighted profiles, and for majority graphs.

We have shown in Section 6.1 that testing whether a candidate is a fair winner over weighted majority graphs is NP-hard.

Let us now consider possible fair winners. Given an incomplete weighted profile $P$, since every balanced agenda is also an agenda, we have that $Poss(P, r_{FSMV}) \subseteq Poss(P, r_{Schwartz})$. We already know from Theorem 5 that determining $Poss(P, r_{Schwartz})$ is difficult. We will now show that this remains so for $Poss(P, r_{FSMV})$.

**Theorem 12** *Given a fixed number $n$ of candidates with $n \geq 3$, deciding if a candidate is in $Poss(P, r_{FSMV})$ is NP-complete.*

**Proof:** We use the same construction as in the proof of Theorem 5. Given the profile constructed there, the only possible fair agendas in which B wins are those in which A plays C, and (at some later point) B then plays the winner. All the additional candidates will be defeated by A, B and C so can be placed anywhere in the fair agenda. □

The computational complexity of determining necessary fair winners remains an open question, just as is the complexity of computing necessary Schwartz winners. However, we know that, since every balanced agenda is also an agenda, $Nec(P, r_{FSMV}) \subseteq Nec(P, r_{Schwartz})$.

# 7 Related work

Since our paper deals with many different issues (fair winners with complete profiles, possible and necessary winners for sequential majority voting and Schwartz, from incomplete profiles and from incomplete majority graphs), related work spans several streams of research, so we structure this section according to the kind of topic studied. First of all, the fair winner rule is a new voting rule. Although we did not identify its complexity, we conjecture that it is NP-hard, which would add one more rule to the zoo of computationally hard voting rules in good company with Dodgson, Kemeny, Young, Slater, etc. See in particular [7] for a synthesis. We now position our results with respect to three related streams of work, namely voting rules with incomplete preferences, computational aspects of voting trees, and computing winners from incomplete tournaments.

## 7.1 Computing voting rules with incomplete preferences

Our paper gives several results about sequential majority voting and the Schwartz rule when voters' preferences are incomplete. The notions of possible and necessary winners for voting rules, as well as the notions of possible and necessary Condorcet winners, have been introduced by Konczak and Lang [9], and their complexity for most common voting rules has been then thoroughly studied by Xia and Conitzer [19]. We complete the picture by adding results concerning sequential majority voting and Schwartz. Moreover, we consider both the standard case of incomplete profiles and the case of incomplete majority graphs, which lead to approximate sets of possible winners, whose computation however is easier than the computation of possible winners in the standard sense.

The complexity of determining possible and necessary winners have also been considered in Pini *et al.* [14] for STV rule. Moreover, they give a preference elicitation procedure which focuses just on the set of possible winners. The problem of deciding if preference elicitation is over (that is, the problem of determining if the remaining votes can be cast so a given candidate does not win) have been previously investigated also by Conitzer and Sandholm [4, 5, 18] .

## 7.2 Voting trees

In this paper we have considered the computational complexity of determining possible winners for sequential majority voting. A specific case of the possible winner problem consists in considering specific partial profiles where a subset of voters provide complete votes and the other voters provide empty votes. In such situations, determining whether a given candidate $x$ is a possible winner is equivalent to decide whether there is a constructive coalitional manipulation for $x$. Conitzer and Sandholm [3] have investigated the complexity of weighted coalitional manipulation for $SMV(T)$ and found out that the latter problem is in $\mathsf{P}$. We prove that the more general possible winner problem for $SMV(T)$ is $\mathsf{NP}$-complete for weighted voters and 3 candidates or more.

Vu et al. [17] consider the following problem. The input consists of a set of candidates and a collection of pairwise winning probabilities $(i, j)$ representing the probability that candidate $i$ beats candidates $j$ in a pairwise election; the problem (called "tournament schedule control") consists in designing a voting tree satisfying some given conditions (such as balancedness) that maximizes the winning probability of a target candidate. It is shown ([17] , Theorems 1 and 3) that finding a balanced tree maximizing the probability of a candidate winning is $\mathsf{NP}$-hard, even if the pairwise winning probabilities are all in $\{0, 0.5, 1\}$. Fair sequential majority voting considered in Section 2.4 corresponds to the restriction of the model of [17] where winning probabilities are deterministic, and our fair winners correspond to winners for which the maximum winning probability over all balanced trees is 1. Whether the $\mathsf{NP}$-hardness result of [17] still holds in the deterministic case, that is, whether computing FSMV is $\mathsf{NP}$-hard, is still an open problem. Note also that both the winning probability matrix in [17] and our incomplete majority graphs in Section 3.2 express some imperfect information about the outcome of every pairwise election. The $\mathsf{NP}$-hardness results of [17] can therefore put aside (although they are technically different) with our Proposition 2, as well as to Theorem 6 in [20]. Hazon et al. [8] also address the maximization over all voting trees of the probability that a given candidate wins the election, and show that a modified version of the problem of designing a tree such that the probability of a given candidate to win is $\mathsf{NP}$-complete.

| | incomplete weighted profiles (n. of candidates) | incomplete majority graphs | incomplete unweighted profiles |
|---|---|---|---|
| *FSMV* | ? | NP-complete (Th. 10) | ? |
| $Poss(r_{SMV(T)})$ | NP-complete ($\geq$ 3) (Th. 2) | P (Th. 4) | NP-complete [20] |
| $Nec(r_{SMV(T)})$ | coNP-complete ($\geq$ 4) (Th. 3) | P (Th. 4) | coNP-complete [20] |
| $Poss(r_{SMV(T)})$, $T$ balanced | NP-complete ($\geq$ 3) (Th. 11) | P(Th. 4) | NP-complete [20] |
| $Nec(r_{SMV(T)})$, $T$ balanced | ? | P(Th. 4) | coNP-complete [20] |
| $Nec(r_{Schwartz})$ | ? | P (Cor. 1) | ? |
| $Poss(r_{Schwartz})$ | NP-complete ($\geq$ 3) (Th. 5) | P (Cor. 2) | ? |
| $Poss(r_{FSMV})$ | NP-complete ($\geq$ 3) (Th. 12) | ? | ? |

Table 1: Complexity of winner determination.

### 7.3 Incomplete tournaments

Brandt *et al.* [2] generalize several tournament solutions to partial tournaments, and study their complexity. Among these solution concepts, they study the Smith set and the Schwartz set (which are known to coincide for complete tournaments). Therefore, the input is an incomplete tournament, just as we do in our Section 3.2. By Lemma 2 of [2] and our Definition 16), the set of the Smith winners for an incomplete tournament $G$ defined as in [2] coincides with the set of our possible Schwartz winners for $G$. Therefore, our polynomiality result (Corollary 2) is subsumed by their Theorem 5. Note that these two works have been developed independently (and simultaneously – both conference papers were published in 2007), with very different starting points and motivations.

Since the Schwartz set from incomplete tournaments as in [2] is contained in the Smith set (Theorem 1 in [2]), we may wonder whether it coincides with the set of our necessary Schwartz winners. However, this does not hold (it suffices to consider the graph $G$ with four candidates $A$, $B$, $C$, $D$, and the edges $(A, B)$, $(B, C)$ and $(C, A)$: $A$ is in the Schwartz set for $G$ (from Lemma 4 of [2]), but from our Theorem 6, it is not a necessary Schwartz winner.

## 8 Conclusions and future work

We have considered multi-agent settings where agents' preferences may be incomplete and they are aggregated using sequential majority voting. We have addressed various computational issues of determining winners for complete/incomplete profiles and majority graphs.

All our results are summarized in Table 1: in the first row we consider the fair sequential majority rule ($FSMV$), where the agents' preferences are assumed to be complete, while in the next four rows we consider the sequential majority rule for a fixed tree ($SMV(T)$) with incomplete preferences, also in the case of balanced trees. The following three rows regard Schwartz and FSMV rules when the agents' preferences are incomplete.

Using the $FSMV$ rule, a winner is a candidate that wins in at least a balanced agenda; we call it a fair winner. We have given a characterization of fair winners via binomial trees and we have shown that it is difficult to find fair winners, when we assume that the agents express their preferences via a complete weighted majority graph (Line 1 of Table 1). Since it is difficult to find an agenda where a candidate wins, it is also difficult for the chair to manipulate the election, thus our results can be interpreted in terms of difficulty of manipulation by the chair (as in, e.g., [1]). Moreover, since it is difficult to find a balanced agenda where a candidate wins, we have also presented a polynomial way to build an agenda with a bounded level of imbalance where a candidate wins. The problem of finding fair winners, when agents' preferences are expressed via a complete profile, is an open problem that we intend to investigate in the future. We have also analyzed the complexity of determining possible fair winners when the agents' votes are incomplete and weighted, and we have shown that also this problem is hard to solve (Line 8 of Table 1).

We have then considered the sequential majority rule when the tree is fixed, i.e., $SMV(T)$, when the agents' preferences are incomplete (Lines 2, 3, 4, and 5 of Table 1). In such a context, we have shown that, if we assume to know agents' preferences under the form of incomplete majority graphs, it is polynomial to compute both the necessary and the possible winners. However, if we assume that agents' preferences are expressed by an incomplete weighted profile, then possible and necessary winners are difficult to compute. We have also shown that possible winners are difficult to compute even when the agenda is balanced. The same results have been shown also in [20] in the case of incomplete unweighted profiles with an unbounded number of candidates. All these hardness results are very interesting, since they imply that is difficult for the chair to manipulate the election.

We then analyzed the Schwartz rule and the FSMV rule in the case of incomplete preferences (Lines 6, 7, and 8 of Table 1). When the preferences are expressed via an incomplete majority graph, it is easy to find possible and necessary Schwartz winners, i.e., those candidates that are Schwartz winners in some completion or in all the completions of preferences. Instead, when the agents' preferences are expressed via an incomplete weighted profile, it is difficult to know if a candidate is a possible Schwartz winner and this remains hard even if we require that the agenda be a balanced tree, i.e., if we consider the FSMV rule. These two results imply that every candidate has an incentive to propose himself as a candidate in the election, that is desirable. In fact, if it is easy to find the set of possible Schwartz winners, then a candidate that is not in this set may avoid to propose himself as a candidate.

The computational complexity of determining necessary Schwartz winners is related to the problem of testing whether constructive control is possible. Such a problem remains open and needs to be studied further, when we consider incomplete profiles. However, we have shown that, it is polynomial if we consider incomplete majority graphs. Thus, it is polynomial to have an approximation of this set of winners.

An interesting direction for future work is deciding which candidates are most likely to win, which is related to probabilistic approaches to voting theory. We also plan to study other forms of uncertainty in the application of the voting rule, such as uncertain weights in a scoring rule, or a chair who can choose between different voting rules.

# References


[1] J.J. Bartholdi, C.A. Tovey, and M.A. Trick. The computational difficulty of manipulating an election. *Social Choice and Welfare*, 6(3):227–241, 1989.

[2] F. Brandt, F. Fischer, and P. Harrenstein. The computational complexity of choice sets. In *Proceedings of TARK'07*, pages 82–91, 2007.

[3] V. Conitzer and T. Sandholm. Complexity of manipulating an election with few candidates. In *Proceedings of AAAI'02*, pages 314–319, 2002.

[4] V. Conitzer and T. Sandholm. Vote elicitation: complexity and strategy-proofness. In *Proceedings of AAAI'02*, pages 392–397, 2002.

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

[6] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. MIT Press, 1990.

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

[8] N. Hazon, Y. Aumann, S. Kraus, and M. Wooldridge. Evaluation of election outcomes under uncertainty. In *Proceedings of AAMAS'08, Volume 2*, pages 959–966. IFAAMAS, 2008.

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

[10] 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.

[11] J.-F. Laslier. *Tournament solutions and majority voting*. Springer-Verlag, 1997.

[12] H. Moulin. *Axioms of Cooperative Decision Making*. Cambridge University Press, 1991.

[13] C. Papadimitriou and M. Yannakakis. The complexity of restricted spanning tree problems. *Journal of the ACM*, 29(2):285–309, 1982.

[14] 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.

[15] 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.

[16] T. Schwartz. Rationality and the myth of the maximum. *Nous*, 6(2):97–117, 1972.

[17] Thuc Vu, A. Altman, and Y. Shoham. On the complexity of schedule control problems for knockout tournaments. In *Proceedings of AAMAS'09, Volume 1*, pages 225–232. IFAAMAS, 2009.

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

[19] 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.

[20] L. Xia and V. Conitzer. Determining possible and necessary winners under common voting rules given partial orders. *http://www.cs.duke.edu/∼lxia*, 2010.