Winner Determination in Sequential Majority Voting*

J. Lang[1], M. S. Pini[2] , F. Rossi[2], K. B. Venable[2] and T. Walsh[3]
1: IRIT, Toulouse, France
lang@irit.fr
2: Department of Pure and Applied Mathematics, University of Padova, Italy
{mpini,frossi,kvenable}@math.unipd.it
3: NICTA and UNSW Sydney, Australia
Toby.Walsh@nicta.com.au

Abstract

Preferences can be aggregated using voting rules. We consider here the family of rules which perform a sequence of pairwise majority comparisons between two candidates. The winner thus depends on the chosen sequence of comparisons, which can be represented by a binary tree. We address the difficulty of computing candidates that win for some trees, and then introduce and study the notion of *fair winner*, i.e. candidates who win in a *balanced* tree. We then consider the situation where we lack complete informations about preferences, and determine the computational complexity of computing winners in this case.

1 Introduction

When the voters' preferences and the voting rule are fully specified, the computation of the winner is easy for many voting rules. Often, however, the voters' preferences or the voting rule itself may be incompletely specified. For instance, only some voters might have expressed their preferences. Even if all voters have expressed their preferences, a new candidates might be introduced. As a third example, we might be eliciting preferences using a compact language such as CP-nets [Boutilier *et al.*, 2004] which induces only a partial or incomplete ordering. [Konczak and Lang, 2005] have therefore considered voting with incompletely specified preferences. They compute the candidates winning in some (resp. all) of the complete extensions of the partial preference profiles for a given voting rule. A second kind of incompleteness is in the voting rule itself. For example, the order in which candidates are compared may not yet be fixed. This can make manipulation by a coalition of voters more difficult [Conitzer and Sandholm, 2003].

In this paper, we consider a well-known family of voting rules based on *sequential majority comparisons*, where the winner is computed from a series of majority comparisons along a binary tree. We study the impact of these two kinds of incompleteness on such voting rules.

The paper is structured as follows. In Section 2 we recall some basics on sequential majority voting. In Section 3 we deal with incompleteness in the voting rule. We study the complexity of computing candidates that win in some or all possible binary trees. Then, in Section 4 we focus on binary trees where the number of competitions for each candidate is as balanced as possible and give characterizations of winners in such trees. In particular, it is possible to build in polynomial time a tree featuring a bounded level of imbalance where a particular candidate A wins, if such a tree exists. Finally, in Section 5 we study the scenario where the agents have only partially revealed their preferences, giving characterizations for notions of possible and Condorcet winners and proving that such winners can be found in polynomial time.

2 Background

Preferences and profiles. We assume that each agent's preferences are specified by a (possibly incomplete) total order (TO) (that is, by an asymmetric, irreflexive and transitive order) over a set of candidates (denoted by Ω). Given two candidates, $A, B \in \Omega$, an agent will specify exactly one of the following: $A < B$, $A > B$, or $A?B$ where $A?B$ means that the relation between A and B has not yet been revealed. A *profile* is a sequence of total orders describing the preferences for a sequence of n agents. An *incomplete profile* is a sequence in which one or more of the preference relations is incomplete. *For the sake of simplicity we assume that the number n of voters is odd.* A Condorcet winner (if it exists) is a candidate A such that for any candidate $B \neq A$, a strict majority of voters prefers A to B.

The majority graph. Given a (complete) profile P, *the majority graph $M(P)$* is the graph whose set of vertices is the set of the candidates Ω and in which 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 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 Ω [Laslier, 1997].

The *weighted majority graph* associated with a complete profile P is the graph $M_W(P)$ whose set of vertices is Ω 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. Weights can quantify the amount of disagreement (e.g. the

*This work has been supported by Italian MIUR PRIN project "Constraints and Preferences" (n. 2005015491).

number of voters preferring A to B). When we want to use standard majority graphs, we just consider all weights to be equal, and we call them simply majority graphs.

Example 1. The majority graph induced by the 3-voter profile $((A > B > C), (B > C > A), (B > A > C))$ has the three edges $B >_m A$, $B >_m C$, and $A >_m C$.

Binary voting trees. Given a set of candidates Ω, a *binary (voting) tree* T is a binary tree where each internal node (including the root) has two children, each node is labelled by a candidate (element of Ω), and the leaves contain every candidate in Ω (one per leaf). Given an internal node n and its two children n_1 and n_2, the candidate associated to n is the winner of the competition between the candidates associated to n_1 and n_2 [Moulin, 1988].

A binary tree T is *balanced* iff the difference between the maximum and the minimum depth among the leaves is less than or equal to 1. In general, such a difference denotes the level of imbalance of the tree.

Sequential majority voting rules induced by binary trees. Given a binary voting tree T, the voting rule induced by T maps each tournament G to the candidate returned from the following procedure (called a *knock-out competition*):

1. Pick a nonterminal node x in T whose successors p, q are terminal nodes; let P and Q be the candidates associated to p and q respectively.
2. Delete the branches $x \rightarrow p$ and $x \rightarrow q$ from T, thus making x a terminal node with candidate P (resp. Q) if $P >_m Q$ (resp. $Q >_m P$) is in G.
3. Repeat this operation until T consists of a single node. Then the candidate associated to this node is returned.

Example 2. Given the majority graph in Figure 1 (a), Figure 1 (b) shows the voting tree corresponding to the sequence of competitions $((A, C), (B, A))$.□

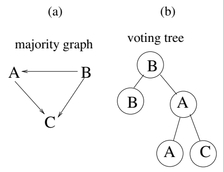

Figure 1: Majority graph and the resulting voting tree.

Possible winners and Condorcet winners. A candidate A is a *Condorcet winner* if and only if, for any other candidate B, we have $A >_m B$. Thus, a Condorcet winner corresponds to a vertex of the majority graph with outgoing edges only. There are profiles for which no Condorcet winner exists. However, when a Condorcet winner exists, it is unique. If there is a Condorcet winner, then it is the sequential majority winner for each binary tree. If there exists at least one binary tree for which a candidate A is the winner, then A is called a *possible winner*.

Many other ways of computing winners from a majority graph exist (see [Laslier, 1997] for a review). For example, the set of possible winners contains the uncovered set [Miller, 1980], which itself contains the Banks set [Banks, 1995]. It also contains the set of Copeland winners, defined as the candidates who maximize the number of outgoing edges in the majority graph. When a Condorcet winner A exists, all these sets coincide with the singleton $\{A\}$.

3 Computing possible winners

The set of possible winners coincides with the *top cycle* of the majority graph [Moulin, 1988]. The top cycle of a majority graph G is the set of maximal elements of the reflexive and transitive closure G^* of G. An equivalent characterization of possible winners is in terms of paths in the majority graph.

Theorem 1 (see e.g. [Moulin, 1988; Laslier, 1997]) *Given a complete majority graph G, a candidate A is a possible winner iff for every other candidate C, there exists a path $B_i \rightarrow B_{i+1}$ for $i = 1, \ldots, k-1$ from A to C in G.*

Since path finding is polynomial, we get the following corollary:

Theorem 2 *Given a complete majority graph and a candidate A, checking whether A is a possible winner and, if so, finding a tree where A wins, is polynomial .*

Example 3. Assume that, given a majority graph G over candidates A, B_2, B_3, and C, candidate A is a possible winner and that only C beats A. Then, for Theorem 1, there must be a path in G from A to every other candidate. Assume $A \rightarrow B_2 \rightarrow B_3 \rightarrow C$ is a path from A to C in G. Figure 2 shows the voting tree corresponding to such a path. □

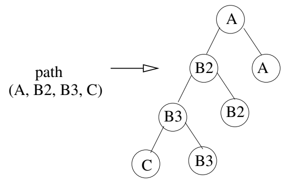

Figure 2: A path and the corresponding voting tree.

4 Fair possible winners

Possible winners are candidates who win in at least one voting 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 voting tree, and we will call such winners *fair possible winners*. For simplicity, we will assume there are 2^k candidates but results can easily be lifted to situations where the number of candidates is not a power of 2. Notice that a Condorcet winner is a fair possible winner, since it wins in all trees, thus also in balanced ones.

We will show that a candidate is a fair possible winner when the nodes of the majority graph can be covered by a binomial tree [Cormen *et al.*, 1990], which means that the nodes of the majority graph are the terminal nodes of a balanced competition tree. 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 a binomial tree of order k with root A. In this situation, we have a balanced voting tree where A wins. Thus A is a fair possible winner.

Example 4. Consider the majority graph G over candidates A, B, C and D depicted in Figure 3. Since such a majority graph is covered by the binomial tree T_2 with root A, we can conclude that A is a fair possible winner. □

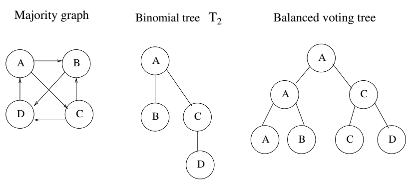

Figure 3: From a majority graph to a balanced voting tree via a binomial tree.

Theorem 3 *Given a complete 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 possible 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 voting 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. □

Notice that a Condorcet winner is a fair winner, and that a fair winner is a possible winner. Moreover, there is always at least one fair winner, thus a voting rule accepting only fair possible winners is well-defined.

Unfortunately, the complexity of deciding if A is a fair possible winner is an open problem. However, for *weighted* majority graphs, it is NP-hard to check whether a candidate A is a fair possible winner. This means that, if we restrict the voting trees only to balanced ones, it is difficult for the chair (if he can choose the voting tree) to manipulate the election. The problem of manipulation by the chair (also called "control") has been first considered in [Bartholdi *et al.*, 1992].

Theorem 4 *Given a complete oriented and weighted majority graph G and a candidate A, it is $\mathcal{NP}$-complete to check whether A is a fair possible winner for G.*

Proof: (Sketch) 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 [Papadimitriou and Yannakakis, 1982] 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[1]. 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, given a possible winner A, 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. If $\Delta(A) = m - 1$, then A is a Condorcet winner, and vice-versa. In fact, this means that there is a tree where A competes against everybody else, and wins. This can be seen as an alternative characterization of Condorcet winners.

It is important to be able to compute $\Delta(A)$. This is an easy task. In fact, once we know that a candidate A is a possible winner, $\Delta(A)$ coicides with the number of outgoing edges from A in the majority graph.

Theorem 5 *Given a majority graph G and a possible winner A, $\Delta(A)$ is the number of outgoing edges from A in G.*

Proof: If A has k outgoing edges, no voting 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 $\Delta(A) \leq k$.

[1] We thank Claude-Guy Quimper for this observation and for suggesting we look at [Papadimitriou and Yannakakis, 1982].

Moreover, it is possible to build a voting 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 $C \rightarrow A$ in 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 $A \rightarrow B_1 \rightarrow B_2 \ldots \rightarrow B_h \rightarrow C$. 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 $B_j \rightarrow \ldots \rightarrow C$ 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 voting tree in which A wins and has depth exactly k. Thus $\Delta(A) = k$. □

We now show that, if A is a possible winner, there exists a voting 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.

Theorem 6 *Given a complete majority graph G and a possible winner A, there is a voting 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.*

Proof: Since A beats $\Delta(A)$ candidates, we can easily build a balanced voting 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 possible 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. □

If unfair tournaments are undesirable, we can consider those possible winners for which there are voting trees which are as balanced as possible. Theorem 6 helps us in this respect: if A is a possible 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), we can give a smaller upper bound on the amount of imbalance in the fairest/most balanced tree in which it wins. Notice that that every Copeland winner is a possible winner.

Theorem 7 *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 Theorem 6 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. □

5 Incomplete majority graphs

Up till now, agents have defined all their preferences over candidates, thus the majority graph is complete; uncertainty comes only from the tree, i.e., from the voting rule itself. Another source of uncertainty is that the agents' preferences may only be partially known.

Let P be an incomplete profile, i.e., a collection of n incomplete preference relations over Ω. The *incomplete majority graph* $M(P)$ induced by P is defined as the graph whose set of vertices is Ω and containing an edge from A to B if the number of voters who prefer A to B is greater than $n/2$. $M(P)$ is thus an incomplete asymmetric graph (or partial tournament) over Ω. For instance, the graph induced by the 3-voter incomplete profile $(A > B > C), (A > C), (A > B, C)$ is the graph with the two edges $A >_m B$ and $A >_m C$.

Note, importantly, that the set of all (complete) majority graphs extending $M(P)$ is a *superset* of the set of majority graphs induced by all possible completions of P; this inclusion can be strict, which means that summarizing P into $M(P)$ implies a loss of information.

We would like to reason about possible winners and Condorcet winners in such a scenario. Following [Konczak and Lang, 2005], we say that $x \in \Omega$ is a $\forall$-possible winner (resp. $\exists$-possible winner) for P iff it is a possible winner in all completions (resp. in some completion) of P, and that $x \in \Omega$ is a $\forall$-Condorcet winner (resp. $\exists$-Condorcet winner) for P iff it is a Condorcet winner in all completions (resp. in some completion) of P. We know from [Konczak and Lang, 2005] that $\forall$-Condorcet winner and $\exists$-Condorcet winners can be computed from P in polynomial time. However, computing $\forall$-possible winners and $\exists$-possible winners looks far less easy (we conjecture that it is NP-hard). Now, by reasoning from the incomplete majority graph rather than from the completions of P, we can define the following four *approximations* of $\exists$- and $\forall$-possible (and Condorcet) winners, which, as we will show shortly, are all computable in polynomial time.

Definition 1 *Let G be an incomplete majority graph and A a candidate.*

- *A is a* weak possible winner *for G iff there exists a completion of G and a tree for which A wins.*
- *A is a* strong possible winner *for G iff for every completion of G there is a tree for which A wins.*
- *A is a* weak Condorcet winner *for G iff there is a completion of G for which A is a Condorcet winner.*
- *A is a* strong Condorcet winner *for G iff for every completion of G, A is a Condorcet winner.*

When the majority graph is complete, strong and weak

Condorcet winners (resp. strong and weak possible winners) coincide, and coincide also with Condorcet winners (resp. possible winners as defined in Section 2).

We denote by $WP(G)$, $SP(G)$, $WC(G)$ and $SC(G)$ the sets of, respectively, weak possible winners, strong possible winners, weak Condorcet winners and strong Condorcet winners for G. We have the following inclusions:

$$SC(G) \subseteq WC(G) \cap SP(G)$$
$$WC(G) \cup SP(G) \subseteq WP(G)$$

Furthermore, because the set of all complete majority graphs extending $M(P)$ contains the set of majority graphs induced by the completions of P, any strong possible winner for $M(P)$ is a $\forall$-possible winner and any $\exists$-possible winner for P is a weak possible winner for $M(P)$. (And similarly for weak/strong Condorcet winners.)

We now give graph-theoretic characterizations for each of the four notions above.

Theorem 8 *Given an incomplete majority graph G and a candidate A, A is a strong possible winner 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 possible winner in every completion of G, i.e., it is a strong possible 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 Ω: $R(A)$ 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 Ω. 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 possible 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 possible winner. This shows that A is not a strong possible winner for G. $\Box$

Clearly, a procedure based on the previous theorem gives us a polynomial algorithm to find strong possible winners.

Let G be an asymmetric graph, Ω the set of candidates, and $A \in \Omega$. Let us consider the following algorithm:

begin
$\Sigma := \{A\} \cup \{X |$ there is a path from A to X in $G\}$;
$G' := G$;
Repeat
  for all $(Y, Z) \in \Sigma \times (\Omega \setminus \Sigma)$ **do**
    if $(Z \rightarrow Y) \notin G'$ **then** add $(Y \rightarrow Z)$ to G';
  for all $Z \in \Omega \setminus \Sigma$ **do**
    if there is a path from A to Z in G'
    then add Z to Σ
Until $\Sigma = \Omega$ or there is no $(Y, Z) \in \Sigma \times (\Omega \setminus \Sigma)$ s. t. $(Z \rightarrow Y) \in G'$;
Return Σ.

Let us call $f(G, A)$ the set Σ returned by the algorithm on G and A. Then we have the following result:

Theorem 9 *$f(G, A) = \Omega$ if and only if A is a weak possible winner for G.*

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 $Y \rightarrow Z$ is added to G' when $Z \rightarrow Y$ is not already in G'.

Now, assume $f(G, A) = \Omega$. 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 possible winner in G'' and therefore a weak possible winner for G. Conversely, assume $f(G, A) = \Sigma \neq \Omega$. Denote $\Theta = \Omega \setminus \Sigma$. Then, for all $(Y, Z) \in \Sigma \times \Theta$ we have $Z \rightarrow Y \in G'$. Now, $Z \in \Theta$ means that no edge $Z \rightarrow Y$ (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 $Z \rightarrow Y \in G'$ if and only if $Z \rightarrow Y \in G$. This implies that for all $(Y, Z) \in \Sigma \times \Theta$ we have $Z \rightarrow Y \in G$, therefore, in every tournament G'' extending G, every candidate of Θ beats every candidate of Σ, 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 possible winner in G''. Since the latter holds for every tournament G'' extending G, A is not a weak possible winner for G. $\Box$

Since the above algorithm for copmuting $f(G, A)$ runs in time $O(|\Omega|^2)$, we get as a corollary that weak possible winners can be computed in polynomial time.

Given an asymmetric graph G, Θ 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 weak possible winners:

Theorem 10 *A is a weak possible winner with respect to G if and only if A belongs to all dominant subsets of G.*

Proof: Suppose there exists a dominant subset Θ 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 weak possible winner for G. Conversely, suppose that A is not a weak possible winner for G. Then the algorithm for computing $f(G, A)$ stops with $f(G, A) \neq \Omega$ and $\Omega \setminus f(G, A)$ being a dominant subset of G. Since $A \in f(G, A)$, $\Omega \setminus f(G, A)$ is a dominant subset of G to which A does not belong. $\Box$

As a consequence of Theorems 8 and 10, weak and strong possible winners are computable in polynomial time, which means that *they can be seen as polynomial approximations of $\exists$-possible and $\forall$-possible winners, respectively* (the computation of those looks much harder). We do not need such a polynomial characterization for $\exists$-Condorcet and $\forall$-Condorcet winners, since these are computable in polynomial time; however, for the sake of completeness we give below simple (and obvious) graph-theoretic characterizations of weak and strong Condorcet winners.

Theorem 11 *Given an incomplete majority graph G and a candidate A, A is the strong Condorcet winner iff A has $m-1$ outgoing edges in G.*

Theorem 12 *Given an incomplete majority graph G and a candidate A, A is a weak Condorcet winner iff A has no ingoing edges in G.*

Given an incomplete majority graph, the set of weak/strong Condorcet winners can therefore be computed in polynomial time from the majority graph.

We end this section by giving bounds on the number of weak/strong possible/Condorcet winners.

Theorem 13 *Let $|\Omega| = m$. The following inequalities hold, and for each of them the bounds are reached: $0 \leq |SC(G)| \leq 1$; $0 \leq |WC(G)| \leq m$; $0 \leq |SP(G)| \leq m$; $1 \leq |WP(G)| \leq m$.*

6 Conclusions

We have addressed various computational issues for sequential majority voting. We summarize here our results and their potential impact.

We first dealt with uncertainty about the choice of the binary tree. Because the choice of the tree is under the control of the chair, our results can be interpreted in terms of difficulty of manipulation by the chair (as in, e.g., [Bartholdi *et al.*, 1992]). We proved that sequential majority voting is easy to manipulate. We gave characterizations of fair possible winners, i.e. possible winners who win in a balanced voting tree.

We also dealt with uncertainty about voters' preferences. We showed that in this case, it is easy to compute a lower and an upper bound of the set of candidates winning for some binary tree. These results apply to manipulation by coalitions of voters and elicitation by the chair as these are two situations where we have to reason with incomplete preferences. See [Konczak and Lang, 2005] for a discussion on the relationship between computing weak/strong winners and computing constructive and destructive manipulations by coalitions of voters [Conitzer and Sandholm, 2002a] and vote elicitation [Conitzer and Sandholm, 2002b].

Many issues remain open, such as the complexity of computing fair possible winners, and the complexity of deciding whether a candidate is a possible winner for some/all completions of an incomplete preference profile.

Acknowledgements

We thank the referees for their many helpful suggestions and corrections. Especially, we are very indebted to one of them, who pointed out to us an important problem in the initial version of the paper and gave us useful suggestions for coping with it. National ICT Australia is funded by the Australian Government's *Backing Australia's Ability* initiative, in part through the Australian Research Council.

References

[Banks, 1995] J. S. Banks. Sophisticated voting outcomes and agenda control. *Social Choice and Welfare*, 1(4):295–306, 1995.

[Bartholdi *et al.*, 1992] J. Bartholdi III, C. Tovey, and M. Trick. How hard is it to control an election? *Mathematical and Computer Modeling*, 16(8/9): 27–40, 1992.

[Boutilier *et al.*, 2004] C. Boutilier, R. I. Brafman, C. Domshlak, H. Hoos, and D. Poole. Preference-based constrained optimization with CP-nets. *Computational Intelligence*, 20(2):137–157, 2004.

[Conitzer and Sandholm, 2002a] V. Conitzer and T. Sandholm. Complexity of manipulating elections with few candidates. In *Proc. of AAAI-02*, pages 314–319, 2002.

[Conitzer and Sandholm, 2002b] V. Conitzer and T. Sandholm. Vote elicitation: complexity and strategy-proofness. In *Proc. of AAAI-02*, pages 392–397, 2002.

[Conitzer and Sandholm, 2003] V. Conitzer and T. Sandholm. Universal voting protocols tweaks to make manipulation hard. In *Proc. of IJCAI-03*, pages 781–788, 2003.

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

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

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

[Miller, 1980] N. Miller. A new solution set for tournaments and majority voting: further graph-theoretical approaches to the theory of voting. *American Journal of Political Science*, 24:68–9, 1980.

[Moulin, 1988] H. Moulin. *Axioms of Cooperative Decision Making*. Cambridge University Press, 1988.

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