Advances in Preference Handling

Although preferences have traditionally been studied in fields such as economic decision making, social choice theory, and Operations Research, they have nowadays found significant interest in computational fields such as Artificial Intelligence, Databases, and Human-computer interaction. This broadened scope of preferences leads to new types of preference models, new problems for applying preference structures, and new kinds of benefits. Explicit preference modelling provides a declarative way to choose among alternatives, whether these are solutions of problems to solve, answers of data-base queries, decisions of a computational agent, plans of a robot, and so on. Preference-based systems allow finer-grained control over computation and new ways of interactivity, and therefore provide more satisfactory results and outcomes. Preference models may also provide a clean understanding, analysis, and validation of heuristic knowledge used in existing systems such as heuristic orderings, dominance rules, and heuristic rules. Preferences are studied in many areas of Artificial Intelligence such as knowledge representation, multi-agent systems, constraint satisfaction, decision making, decision-theoretic planning, and beyond. Preferences are inherently a multi-disciplinary topic, of interest to economists, computer scientists, operations researchers, mathematicians and more.

This workshop is intended as a multidisciplinary event that brings together researchers from these different fields and allows them to exchange experiences and to discuss advanced methods for preference handling. It thus continues a series of multidisciplinary workshops on preference handling (a AAAI-02 workshop, a Dagstuhl-Seminar in 2004, and an IJCAI-05 workshop) which have been all very successful.

We have received 32 submissions to this workshop which confirms a continued interest in research on preference handling. The program committee has selected 21 papers for the two-day workshop. The selected papers do not only cover advances in theoretical topics such as preference elicitation, preference representation, preference aggregation, voting theory, multi-criteria optimization, data-base queries, but also interesting applications in areas such as mechanism design, planning and reasoning about actions, and recommender systems.

We welcome all participants of the multidisciplinary ECAI 2006 workshop on advances in preference handling and hope that this event will not only stimulate new ideas and insights in preference handling, but also lead to a broader understanding of the possible applications of this emerging domain.

Ulrich Junker
Werner Kießling

August 2006

Organization

Program Chairs

Ulrich Junker, ILOG, France
Werner Kießling, University of Augsburg, Germany

Program Committee

Elisabeth André, University of Augsburg, Germany
Wolf Tilo Balke, University of Hannover, Germany
Ronen Brafman, Stanford University, USA
Gerhard Brewka, University of Leipzig, Germany
Kevin C. Chang, University of Urbana-Champaign, USA
Jan Chomicki, University at Buffalo, USA
Paolo Ciaccia, University of Bologna, Italy
Carmel Domshlak, Technion - Israel Institute of Technology, Israel
Jon Doyle, North Carolina State University, USA
Matthias Ehrgott, University of Auckland, New Zealand
Parke Godfrey, York University, Canada
Judy Goldsmith, University of Kentucky, USA
Eric Horvitz, Microsoft Research, USA
Vagelis Hristidis, Florida International University, USA
Ihab Ilyas, University of Waterloo, Canada
Ulrich Junker, ILOG, France
Werner Kießling, University of Augsburg, Germany
Jerome Lang, IRIT - Univ. Paul Sabatier, France
Barry O'Sullivan, University College Cork, Ireland
Simon Parsons, Brooklyn College, USA
Patrice Perny, LIP6 - Paris 6 University, France
Pearl Pu, EPFL, Switzerland
Francesca Rossi, University of Padova, Italy
Alexis Tsoukiàs, LAMSADE, France

Winner Determination in Sequential Majority Voting with Incomplete Preferences

J. Lang[1] **M.S. Pini**[2] **F. Rossi**[3] **K. Venable**[4] and **T. Walsh** [5]

Abstract. Voting rules map collections of preference orderings over a set of candidates (one for each voter) to candidates. Now, in many contexts, we have to consider the case where either the voters' preferences or the voting rule itself are incompletely specified. We consider here the family of voting rules consists of sequential majority comparisons, where the winner is computed from a series of majority comparisons along a binary tree. We address the computation of the winner of such voting rules, when the preferences and/or the binary tree are incompletely specified.

1 INTRODUCTION

Voting rules map collections of preference orderings over a set of candidates (one for each voter) to candidates. When the voter's preferences and the voting rule are fully specified, the computation of the winner is generally easy (except for a few voting rules, see e.g. [2, 7, 12]). Now, in many contexts, we have to consider the case where either the voters' preferences or the voting rule itself are incompletely specified.

Considering the application of voting rules to *incomplete preferences* is particularly relevant in the following situations:

- some voters have expressed their preference profile and some others have not yet done it; in that case, the collective preference profile is a collection consisting of n_1 complete preference relations and $n - n_1$ empty preference relations.
- all voters have expressed their preferences on a given subset of candidates, and now new candidates are introduced, about which the voters' preferences are unknown.
- voters are allowed to express their preferences in an incremental way: they left some comparisons between candidates unspecified, because either they *don't know* or they *don't want* to compare some candidates (we comment further on the various possible interpretations of incomplete preferences).
- preferences have been only partially elicited and/or are expressed in a language for compact preference representation such as CP-nets [3] which induce partial preference relations in the general case.

Voting with incomplete preferences was considered in [8], which addressed the computation of the candidates winning in some (resp. all) of the complete extensions of the partial preference profiles (for a given voting rule). A closely related issue is *vote elicitation* [4], which consists in determining, given a set of individual profiles for a subset of voters who have already expressed their votes, whether (a) the outcome of the vote can be determined without needing any further information and (b) which information must be asked to which voter.

Considering *partially specified voting rules* is particularly relevant for making manipulation by coalitions of voters more difficult (see for instance [5] where the introduction of some uncertainty in the application of voting rules makes manipulations computationally harder). They are also relevant for addressing the issue of the manipulation of an election by the chairman, since the latter may have the power of removing the uncertainty on the voting rule.

A well-known family of voting rules consists of *sequential majority comparisons*, where the winner is computed from a series of majority comparisons along a binary tree.

In this paper we study the impact of the above two kinds of incompleteness on the application of such voting rules. Incompleteness of the voters' preferences leads to applying such sequential majority comparisons to an *incomplete majority graph*. Incompleteness of the voting rule itself leads to applying sequential majority comparisons with an unknown (or partially defined) tree.

In Section 2 we recall some basics on voting theory and sequential majority comparisons. In Section 3 we deal with incompleteness in the voting rule. We study the computational difficulty of computing candidates that win for any some binary tree (called *possible winners*), and we notice that a candidate win for every binary tree if and only if it is a Condorcet winner. Then, in Section 4 we focus on a particular subclass of binary trees: we are interested only in fair sequences, where the number of competitions for each candidate is as balanced as possible. We show that, although winner determination in this context looks hard, it is however possible, for a given candidate A, to build in polynomial time a tree featuring a bounded level of imbalance where A wins, if such a tree exists. In Section 5 we cope with the other form of uncertainty, occurring when the agents reveal their preferences only partially. In this new scenario we extend the notions of possible winners and Condorcet winners to those of weak and strong possible/Condorcet winners, and we address computational issues for the determination of these winners.

2 BACKGROUND

Preferences and profiles. We assume that each agent's preferences are specified via a (possibly incomplete) total order (TO) over the set of possible candidates, that we will denote by Ω. Given two candidates A and B, an agent will specify exactly one of the follow-

[1] IRIT, Toulouse, France , email:lang@irit.fr
[2] University of Padova, Italy, email: mpini@math.unipd.it
[3] University of Padova, Italy, email: frossi@math.unipd.it
[4] University of Padova, Italy, email: kvenable@math.unipd.it
[5] NICTA and UNSW Sydney, Australia, email:Toby.Walsh@nicta.com.au

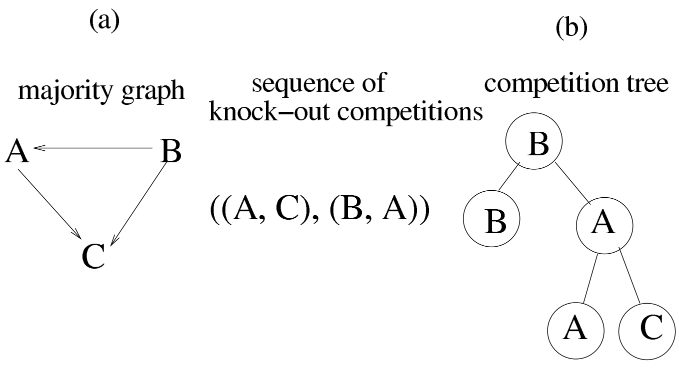

Figure 2. Majority graph, sequence of knockout competitions and the resulting competition tree.

graph. When a Condorcet winner A exists, all these sets coincide with the singleton $\{A\}$.

3 COMPUTING POSSIBLE WINNERS FROM A COMPLETE MAJORITY GRAPH

Clearly, determining whether a candidate is a Condorcet winner can be done in time linear in the size of the majority graph (that is, quadratic in the number m of candidates, since the majority graph can be obtained with a time complexity $O(m^2 \times n)$).

A constructive characterization of possible winners is already known in the literature[6]: possible winners coincide with the elements in the *top cycle* of the majority graph (see for instance [11]). (The top cycle of a majority graph G is the set of maximal elements of the reflexive and transitive closure G^* of G). *Although the result is not original*, we give a new proof of it (in particular because it shows how to build a tree for which a given possible winner wins).

Theorem 1 (see e.g. [11, 9]) *Given a complete majority graph G, a candidate A is a possible winner iff for every other candidate C, there exists a path from A to C in G.*

Proof: ($\Rightarrow$) Assume that A is a possible winner and that T is one of the competition trees where A wins.

For every node x of T let us denote with $label(x)$ the candidate represented by x, with $w(x)$ the child of x such that $label(w(x)) = label(x)$ and with $l(x)$ the other child. Notice that in the majority graph there is the edge $label(x) \rightarrow label(l(x))$.

Take any candidate C different from A. We will now show how to find a path from A to C in G.

Consider procedure $P(C, T(x))$ such that T is a tree with root x and C is a candidate, that finds a path from $label(x)$ to C in the majority graph if C is a label of one or more nodes in T. If $C \in T(l(x))$, there exists arc $label(x) \rightarrow label(l(x))$ in G, so procedure P adds this arc to the path and then calls procedure $P(C, T(l(x)))$, else, if $C \in T(w(x))$, P just calls $P(C, T(w(x)))$. The base case for P is when we call $P(C, T(x))$ and $label(l(x)) = C$. In this case P adds arc $label(x) \rightarrow C$ which is in G.

It is sufficient to call $P(C, T)$ for every C different from A to find a path from A to any other candidate.

($\Leftarrow$) On the other hand, if A has a path to all other candidates, then it is a possible winner. In order to show this, we give the following algorithm, called *Tree*, which obtains a tree where A wins, starting from the paths from A to all other candidates.

Algorithm Tree. Consider all paths starting from A of length 2 or more, assuming that all candidates are reachable from A. Take one of such paths, say $p = (A = B_1 \rightarrow B_2 \rightarrow \ldots \rightarrow B_k)$. Consider the trees T_i, for $i = 2, \ldots, k$, defined as follows: $T_k = B_k$; for $i = 2, \ldots, k-1$, $root(T_i) = B_i$, $left(T_i) = B_i$, $right(T_i) = T_{i+1}$. Let us now consider tree T_1, which will be the starting point for the tree T to be built. We will now augment T by considering the other paths from A. For each of them, let us take the latest element B such that B is one node of T. Then we build the tree corresponding to the path portion from B to the end, and we attach it to T where B appears as a leaf. We do the same with the other paths until all of them have been considered. At the end, we have a competition tree rooted at A. We now take all paths from A of length 1, and we add competitions between A and each of these outcomes. Thus we have a competition tree where A is the root and all candidates appear. Q.E.D.

As noted above, possible winners coincide with elements in the top cycle $tc(G)$ of G, defined as the set of maximal elements of the transitive closure G^* of G: $x \in tc(G)$ if for all $y \neq x$, (x, y) is in G^*. This means that, for all $y \neq x$, there is a path from x to y in G, as required by Theorem 1. Thus the characterization given by theorem 1 is equivalent to the top cycle characterization. However, Algorithm Tree provides a constructive way to build a tree where the possible winner wins.

Corollary 1.1 *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 in the number of the candidates.*

Proof: Because path finding in a graph is polynomial, we can check in polynomial time whether A is a possible winner ot not, and if it is, then a tree can be generated in polynomial time using algorithm $Tree$. In fact, $Tree$ considers at most $m - 1$ paths starting from A, and each path is considered at most once. Q.E.D.

Example 5. 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 3 shows the competition tree obtained applying algorithm $Tree$ to such path.$\Box$

We end this section by two obvious remarks. First, the set of possible winners is a singleton $\{A\}$ if and only if A is a Condorcet winner. Second, if the majority graph G is transitive, then the dominating candidate is a Condorcet winner. Notice that a candidate can be a Condorcet winner even if G is not transitive: consider for example the following majority graph $G = \{A >_m D, A >_m B, A >_m C, B >_m C, C >_m D, D >_m B\}$. In this case A is a Condorcet winner even if there is a cycle connecting the candidates B, C and D.

4 FAIR POSSIBLE WINNERS

Possible winners, as we have defined them, are candidates which win in at least one competition tree. However, such a tree may be very

[6] It is a folklore result, mentioned in many places [11, 9]. We couldn't find the exact place where it first appears.

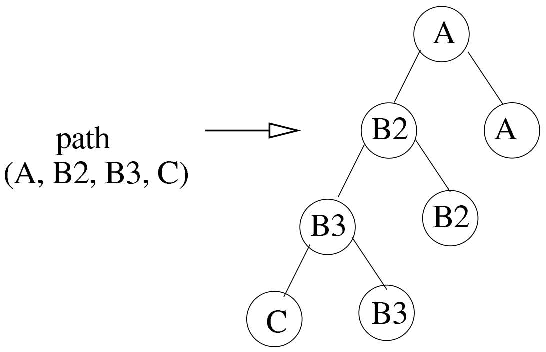

Figure 3. Competition tree corresponding to path $A \to B_2 \to B_3 \to C$ in Example 5.

unbalanced, thus representing a tournament where the winner may compete with few other candidates. This may be considered unfair.

In the following, we will consider a competition fair if it is represented by a balanced competition tree, and we will call such winners *fair possible winners*. 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 tree with a certain shape, i.e. when the nodes of the majority graph are the terminal nodes of a competition tree with a certain shape. Such (non-binary) trees will be called *balancing trees*, and are defined inductively as follows:

- T_1 is the tree with only one node labeled with 1.
- T_k, with $k = 2^i$, is the tree with 2^i nodes built by taking two instances, t_1 and t_2, of tree $T_{2^{i-1}}$, renumbering the labels of the nodes of t_2 by adding a constant equal to 2^{i-1} to their labels, and connecting t_1 and t_2 by setting the root of t_2 as a child of the root of t_1.
- If $2^{i-1} < k \leq 2^i$, then T_k is just T_{2^i} where we remove all nodes labeled with numbers higher than k.

Theorem 2 *Given a complete majority graph G with k nodes, and a candidate A, if there is a balancing tree T_k covering all nodes of G with root A, then A is a fair possible winner.*

Proof: Assume there is a balancing tree T_k satisfying the statement of the theorem. We will show that it is possible to define, starting from T_k, a balanced competition tree B_k where A wins. The definition of B_k is given by induction:

- B_1 coincides with the node covered by the node of T_k with label 1.
- B_k, with $k = 2^i$, is the tree with 2^i leaves built defining two instances of the balanced trees $B_{2^{i-1}}$, b_1 and b_2, obtained renumbering the nodes of b_2 by adding 2^{i-1} to their number, by setting as the root of B_k the root of T_k and by putting b_1 and b_2 as its children.
- If $2^{i-1} < k \leq 2^i$, then B_k is just B_{2^i} where we remove all nodes with number higher than k. Then for every node x that is the unique child of its father, then we set the left child of x as the left child of the father of x and the right child of x as the the right child of the father of x and then we remove x.

Q.E.D.

Example 6. Consider the majority graph G over candidates A, B, C and D depicted in Figure 4. Since such majority graph is covered by balancing tree T_4 with root A, we can conclude that A is a fair possible winner.□

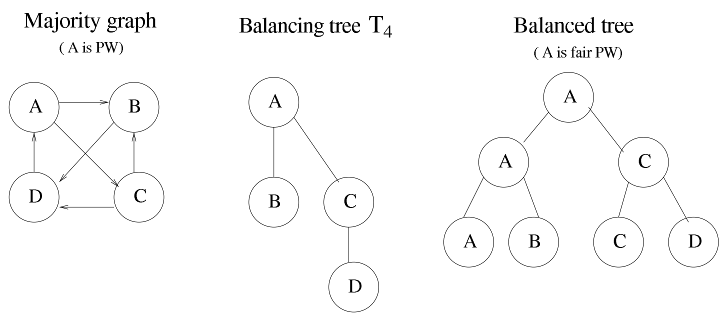

Figure 4. From a majority graph to a balanced tree via balancing tree.

Notice that the set of possible winners contains the set of fair possible winners, which in turn contains the set of Condorcet winners. However, while there could be no Condorcet winner, there is always at least one fair possible winner (because there exist balancing trees). Thus, a voting rule accepting only fair possible winners is well-defined.

Based on the conjecture that finding fair possible winners is difficult, we propose a way to find in polynomial time, given a possible winner A, a tree, with a bounded level of imbalance, in which A wins.

Given a candidate A and a competition tree T rooted at A, we define $D(T, A)$ as the length of the path in the tree T from the root to the only leaf labelled A. Moreover, we call $\Delta(A)$ the maximum $D(T, A)$ over all competition trees T where A wins.

Given a possible winner A, we will now show that $\Delta(A)$ is related to the possibility for A of being a Condorcet winner, or to win in a fair competition. If $\Delta(A) = m - 1$, then A is a Condorcet winner, and viceversa. 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. Moreover, we will now show that, if A is a possible winner, there exists a competition tree where A wins with level of imbalance at most $m - \Delta(A) - 1$. Notice that there could also be trees, in which A wins, more balanced than this one.

Theorem 3 *Given a complete majority graph G and a possible winner A, there is a competition tree with level of imbalance smaller than or equal to $m - \Delta(A) - 1$, where A wins.*

Proof: Since A beats $\Delta(A)$ candidates, we can easily build a balanced competition 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. Thus applying algorithm $Tree$ to BT and such remaining candidates, in the worst case, we will add a subtree of depth $m - \Delta(A) - 1$ rooted at one of the nodes beaten by A in BT. Q.E.D.

Theorem 4 *Given a complete majority graph G and a possible winner A, we can build in polynomial time a competition tree where A wins with level of imbalance equal to* $m - \Delta(A) - 1$.

Proof: The procedure illustrated in the proof of Theorem 3 is clearly polynomial. In fact BT can be built in linear time in $\Delta(A)$ since A beats every candidate of BT and thus the order of the competitions among the other candidates in BT can be set in any way respecting the balance constraint. Moreover, as noted above, the complexity of $Tree$ is polynomial. Q.E.D.

If we think that unfair tournaments are not desirable, we can consider those possible winners for which there are competition trees which are as balanced as possible. The theorem 3 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. For example, if $\Delta(A) = m - 3$, we know that there is a tree with an imbalance level 2 (since $m - \Delta(A) - 1 = 2$) 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.

It is thus 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 competition 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 competition 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 will now consider, one after the other, each candidate C such that $C \to A$ in G. For each such candidate we will 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_h 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 competition tree, in which A wins and has depth exactly k. Thus $\Delta(A) = k$. Q.E.D.

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.

Theorem 6 *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 balanced tree BT as defined in the proof of Theorem 3 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. Q.E.D.

5 COMPUTING THE WINNERS FROM AN INCOMPLETE MAJORITY GRAPH

In the setting considered so far, agents define 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 comes from the fact that the agents' preferences may only be partially known. In this case, the majority graph can be incomplete. We would like to reason about the winners in such a scenario. Therefore, in this section we assume to have an incomplete majority graph, i.e., a graph where some edges are missing.

To cope with this new setting, we need to extend the notions of possible and Condorcet winners as follows. In particular, we will call a *strong Condorcet winner* a candidate which is a Condorcet winner for all completions of the majority graph. On the other hand, a *weak Condorcet winner* is a Condorcet winner for at least one completion of the majority graph. When the majority graph is complete, strong and weak Condorcet winners coincide, and coincide also with the notion of Condorcet winners given above. Similar notions are *strong possible winners* and *weak possible winners*.

We will then give characterizations of these new kinds of winners that allow us to find their set or an approximation of them in linear or polynomial time.

More formally,

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

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:

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

We now give a characterization for each of the four notions above.

Theorem 7 *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 $Others = \Omega \backslash (R(A) \cup R^{-1}(B))$. 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:

Step 1 $\hat{G} := G$;
Step 2 for all $x \in R(A)$ and all $y \in R^{-1}(B)$, add (y, x) to $\hat{G}$;
Step 3 for all $x \in R(A)$ and all $y \in Others$, add (y, x) to $\hat{G}$;
Step 4 for all $x \in Others$ and all $y \in R^{-1}(B)$, add (y, x) to $\hat{G}$;
Step 5 for all x, y belonging to the same element of the partition: if neither (x, y) nor (y, x) in 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 constructin 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. Q.E.D.

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'
      end if
    end do;
    for all $Z \in \Omega \setminus \Sigma$
    do
      if there is a path from A to Z in G'
      then add Z to Σ
      endif
    end do
  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 8 *$f(G, A) = \Omega$ 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 asymmetric 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. Q.E.D.

Since the algorithm computing $f(G, A)$ 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 9 *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. Q.E.D.

In the following we will characterize the weak/strong Condorcet winner and then we will use this characterization for stating that it is linear to compute the exact set of weak/strong Condorcet winners.

Theorem 10 *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.*

Proof: Follows directly from the fact that, given a majority graph, A is a Condorcet winner iff A has only outgoing edges in G. Q.E.D.

Theorem 11 *Given an incomplete majority graph G and a candidate A, A is a weak Condorcet winner iff A has no ingoing edges in G.*

Proof: If A has no ingoing edges, then it has only outgoing or missing edges. If we replace the missing edges of A with outgoing arcs, and we complete the other missing edges arbitrarly, we have a completion of G such that, for every knock-out competition based on G, A wins. Thus A is weak Condorcet winner.

If there is an ingoing edge to A then for every completion G' of G this ingoing edge is present, therefore A cannot be a Condorcet winner in G'. Q.E.D.

Given an incomplete majority graph, the set of weak/strong Condorcet winners can therefore be computed in polynomial time from the majority graph[7].

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

Theorem 12 *Let $|\Omega| = m$. The following inequalities hold, and for each of them the bounds are reached.*

- $0 \leq |SC(G)| \leq 1$;
- $0 \leq |WC(G)| \leq m$;
- $0 \leq |SP(G)| \leq m$;
- $1 \leq |WP(G)| \leq m$;

Proof: The point for strong Condorcet winner is obvious. For weak Condorcet winners, if G is the empty graph, all candidates are weak Condorcet winners; if G is a complete tournament without a Condorcet winner, there is no weak Condorcet winner. For strong possible winners: if we take a complete graph containing $X_1 > X_2 > \ldots > X_m > X_1$ (whatever the remaining edges), the top cycle is Ω, therefore there are m possible winners. For the lower bound: if we take G to be the empty graph, then for each candidate X it is possible to find a completion of G for which X is not a possible winner, the upper bound m is reached when it is reached for $|SP(G)|$ (or for $|WC(G)|$); the lower bound 1 is obtained for any complete graph with a Condorcet winner. Q.E.D.

6 CONCLUSION AND FUTURE WORK

We have considered scenarios where several agents express their preferences over a set of alternatives and their preferences are aggregated by sequential majority rules. We have investigated how to overcome the problem of determining the winner in this scenario that is affected by the uncertainty of majority voting rule that can lead a non-transitive ordering over candidates and so to a not clear definition of the winner and by the uncertainty derived by the fact that some agents can reveal only partially their preferences.

We have defined particular classes of winners: possible and Condorcet winners in the case of complete information concerning the preferences given by the agents and weak/strong possible and Condorcet winner in the case of incomplete preferences. Then we have found interesting characterizations of these classes that have allowed us to compute them or an approximation of them in linear or polynomial time both assuming to have complete preferences and assuming to have incomplete ones.

Moreover we have characterized another class of possible winners, i.e. the fair possible winners, that are possible winners that can win only in balanced sequences of knockout competitions and we have shown that is polynomial to find for a possible winner the minimum level of imbalance of the trees where it wins.

In the future we want to investigate how the chairman that decides the sequence of knock-out competitions can interfere and manipulate the winner determination both in the case of complete preferences of agents and in the case of incomplete ones.

[7] Notice that, although they seem to coincide at first glance, the notions of weak/strong Condorcet winner and the notions of possible/necessary Condorcet winners in [8] are different, because the input is not the same (incomplete profiles in [8], an incomplete majority graph here), and that summarizing an incomplete profile by an incomplete majority graph can induce a loss of information.

ACKNOWLEDGEMENTS

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

We wish to thank the anonymous referees for many helpful suggestions.

REFERENCES

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

[2] J.J. Bartholdi, C.A. Tovey, and M.A. Trick, 'Voting schemes for which it can be difficult to tell who won the election', *Social Choice and Welfare*, **6**(3), 157–165, (1989).

[3] 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).

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

[5] V. Conitzer and T. Sandholm, 'Universal voting protocols tweaks to make manipulation hard', in *Proceedings of IJCAI-03*, pp. 781–788, (2003).

[6] Marquis de Condorcet, 'Essai de l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix', (1785).

[7] E. Hemaspaandra, L. Hemaspaandra, and J. Rothe, 'Exact analysis of Dodgson elections: Lewis Carroll's 1876 system is complete for parallel access to NP', *JACM*, **44**(6), 806–825, (1997).

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

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

[10] 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).

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

[12] J. Rothe, H. Spakowski, and J. Vogel, 'Exact complexity of the winner for Young elections', *Theory of Computing Systems*, **36**(4), 375–386, (2003).