# Voting with Partial Information: What Questions to Ask?

Ning Ding and Fangzhen Lin

**Abstract**

Voting is a way to aggregate individual voters' preferences. Traditionally a voter's preference is represented by a total order on the set of candidates. However, sometimes one may not have a complete information about a voter's preference, and in this case, can only represent a voter's preference by a partial order. Given this framework, there has been work on computing the possible and necessary winners of a (partial) vote. In this paper, we take a step further, look at sets of questions to ask in order to determine the outcome of such a vote. Specifically, we call a set of questions a deciding set for a candidate if the outcome of the vote for the candidate is determined no matter how the questions are answered by the voters, and a possible winning (losing) set if there is a way to answer these questions to make the candidate a winner (loser) of the vote. We discuss some interesting properties about these sets of queries and prove some complexity results about them under some well-known voting rules such as plurality and Borda.

## 1 Introduction

Voting is a general way to aggregate preferences when a group of people need to make a common decision but have disagreements on which decision to take. Voting is traditionally studied in game theory and social choice theory. Recently it has attracted much attention in AI for various reasons, see for example the survey [Chevaleyre *et al.*, 2007].

Traditionally, a voter's preference is assumed to be a complete linear order over possible candidates (outcomes,or alternatives). One can easily imagine situations where this assumption is too strong, either because the voter herself cannot rank all of the possibilities linearly or because as an observer, we do not have a complete knowledge about her preferences. In fact, one of the well-known formalisms for representing agents' preferences in AI, CP-nets [Boutilier *et al.*, 2004], assumes agents' preferences are partial-ordered. In the context of voting, there has been work in this direction as well. Given a partial ordering for each voter, Konczak and Lang [2005] considered the problem of deciding whether a candidate is a *necessary winner* and *possible winner*. A necessary winner is a candidate who is always a winner in every possible completion of the given partial preference profile, while a possible winner is one who is a winner in some of the completions. The complexities of these two problems under a variety of voting rules, especially the so-called positional scoring rules, have been extensively studied [Pini *et al.*, 2007; Xia and Conitzer, 2011; Betzler and Britta, 2010; Baumeister and Rothe, 2010]. More recently, Conitzer *et al.* [2011] considered a notion of manipulations in voting with partial information.

In this paper, we continue this line of work. Given a voting context consisting of a set of candidates, a set of voters, and for each voter, a partial order on the candidates, we consider in general how much additional information is still needed in order to make a particular candidate a winner or loser under a voting rule. If the candidate is already a necessary winner or a necessary loser, then no additional information is needed. Otherwise, one may want to know which voter is crucial in deciding the outcome, and for that voter what would be the important questions to ask. These are obviously important issues to consider when doing voter preference solicitation, and should have some interesting applications. For instance, in an election, a candidate's team may want to know that given what they already know about a group of people, whether more knowledge about their voting preferences would make any differences to the outcome of the election.

This "additional information or knowledge" can come in many forms. Here we take it to be a set of pair-wise comparison questions [Conitzer, 2009] of the following form: voter $i$, which candidate do you prefer, $a$ or $b$? We then consider sets of these questions that can settle the outcome for a candidate. There are at least two possible approaches here. A cautious approach looks for a set of questions such that no matter how these questions are answered by the voters will determine whether the candidate will win or lose. We call such a set of queries a deciding set for the candidate. This amounts to saying that as far as the candidate is concerned, if a question is not in a deciding set, then this question is irrelevant and can be ignored. It is thus not surprising that there is a unique minimal deciding set regardless of which voting rule to use.

The cautious approach makes sense when we want additional information that can decide the outcome for the candidate in question. If we want additional information that would make the candidate a winner (or a loser), then another notion may be more appropriate. Consider the case when we want a candidate to be a winner. Here we may be interested in a set of questions for which there are answers that would lead to the candidate being a winner (or loser), hoping that when voters are asked about these questions they will either indeed answer them as expected or that we can somehow influence them to answer them that way. We call such a set of questions a possible winning (or losing) set for the candidate. As can be expected, minimal possible winning (or losing) sets may not be unique. In contrast to our static notion of query sets, [Conitzer and Sandholm, 2002] defined the dynamic notion of *elicitation tree* and studied some basic problems related to that concept.

The rest of the paper is organized as follows. We first review some basic notions of voting with complete and partial information, and the notions of possible and necessary winners [Konczak and Lang, 2005]. We then define our notions of deciding sets, possible winning sets, and possible losing sets. We then prove some interesting properties about deciding sets, and consider how to compute minimal deciding set under various votin g rules. We next do the same for possible winning sets, and then conclude the paper.

# 2 Preliminaries

We assume a finite set $N = \{1, ..., n\}$ for voters (players, or agents), and a finite set $O$ for candidates (outcomes, or alternatives). A *preference ordering* $p_i$ of a voter $i$ is a total (linear) order on $O$, and a preference profile $p$ is a tuple of preference orderings, one for each voter.

A voting rule (method) $f$ is a function from preferences profiles to non-empty sets of outcomes. For a preference profile $p$, $f(p)$ is the set of winners. When a single winner is desired, a tie-breaking rule can be used to select the one from $f(p)$. Or $f$ is required to be single-valued. In social choice theory terminology, when $f(p)$ can be a set of outcomes, it is called a social choice correspondence, and when $f(p)$ is always single-valued, it is called a social choice function.

Most of the popular voting rules can be defined using a score vector $(s_1, s_2, \cdots, s_m)$, where $m$ is the number of candidates, and $\forall i < m$, $s_i \geq s_{i+1}$. Given such a score vector, for each voter $i$ and preference ordering $p_i$, the $k$th ranked candidate according to $p_i$ receives the score $s_k$ from the voter. Given a preference profile $p$, a candidate's score is then the sum of of the scores that she receives from each voter, and the winners are those that have the highest score. Such voting rules are called scoring rules.

For instance, the *plurality* voting rule uses the score vector $(1, 0, \cdots, 0)$, the *veto* rule uses the score vector $(1, \cdots, 1, 0)$, and the *Borda* rule uses the score vector $(m, m-1, \cdots, 1)$.

As mentioned in the introduction, we consider the situation when the preference ordering of a voter may not be total, either because the onlooker who is studying the voting does not have a complete knowledge of the voter's preference or that the voter herself is not certain of her own preferences.

Formally a *partial preference ordering* $p_i$ of voter $i$ is a partial order on the set $O$ of candidates:

for each $o \in O$, $(o, o) \in p_i$ (reflexivity), if both $(o_1, o_2)$ and $(o_2, o_1)$ are in $p_i$, then $o_1 = o_2$ (antisymmetry), and if $(o_1, o_2)$ and $(o_2, o_3)$ are in $p_i$, then $(o_1, o_3) \in p_i$ (transitivity). A *partial preference profile* is then a tuple of partial preference orderings, one for each voter.

Given a partial preference ordering $p_i$, an *extension* of $p_i$ is a partial preference ordering $p'_i$ such that $p_i \subseteq p'_i$. An extension of $p_i$ that is a total order is called a *completion* of $p_i$. Similarly, an extension of a partial preference profile $p$ is a partial preference profile $p'$ such that for each $i$, $p'_i$ is an extension of $p_i$, and a completion of a partial preference profile $p$ is a preference profile that is an extension of $p$.

Under a voting rule $f$, a candidate $o$ is said to be a *necessary winner* of a partial preference profile $p$, if for all completion $p'$ of $p$, $o \in f(p')$. If there exists such a completion, then $o$ is said to be a *possible winner* [Konczak and Lang, 2005]. Furthermore, if $o$ is not a possible winner, then we call $o$ a necessary loser; and if $o$ is not a necessary winner, then we call $o$ a possible loser.

# 3 Deciding sets, possible winning sets, and possible losing sets of queries

As mentioned in the introduction, our interest in this paper is on getting additional information to decide the outcome of a vote. This additional information will be in the form of comparison queries [Conitzer, 2009] to voters.

**Definition 1** *A (comparison) query to voter $i$ is one of the form $i$:$\{a, b\}$ that asks $i$ to rank candidates $a$ and $b$.*

When presented with the query $i$:$\{a, b\}$, the voter $i$ has to answer either "a" (she prefers $a$ over $b$) or "b" (she prefers $b$ over $a$).

**Definition 2** *An answer to a set $Q$ of questions is a function $\sigma$ from $Q$ to $O$ such that for any $i$:$\{a, b\} \in Q$, $\sigma(i$:$\{a, b\}) \in \{a, b\}$.*

Intuitively, if an answer $\sigma$ maps $i$:$\{a, b\}$ to "a", then the preference $(a, b)$ ($a \geq b$) is added to voter $i$'s partial preference ordering, and this may entail some new preferences for $i$, and may even lead to a contradiction. In the following, we require an answer to be consistent with the preferences that the voters already have.

**Definition 3** *Let $p$ be a partial preference profile and $Q$ a set of queries. An answer $\sigma$ to $Q$ is legal under $p$ if for each voter $i$, the transitive closure of the following set*

$$p_i \cup \{(a, b) \mid i\text{:}\{a, b\} \in Q \wedge \sigma(i\text{:}\{a, b\}) = a\}$$

*which we denote by $p_i(\sigma, Q)$, is a partial order on $O$, the set of candidates. Given a legal answer $\sigma$ to $Q$ under $p$, the resulting partial preference profile is then*

$$p(\sigma, Q) = (p_1(\sigma, Q), ..., p_n(\sigma, Q)),$$

In the following, unless stated otherwise, we always assume that answers to sets of questions are legal under the given partial preference profile.

We can now define the sets of questions that we are interested in this paper. A deciding set of queries for a candidate $o$ determines the outcome of the vote for $o$ no matter how the queries in the set are answered.

**Definition 4** *Let $p$ be a partial preference profile, $o$ a candidate, and $f$ a voting rule. A set $Q$ of queries is a deciding set for $o$ (in $p$ under $f$) if for every answer $\sigma$, $o$ is either a necessary winner or a necessary loser in the new partial profile $\sigma(p, Q)$ under $f$. $Q$ is a minimal deciding set for $o$ if it is a deciding set and there is no other deciding set $Q'$ such that $Q' \subset Q$.*

Consider the incomplete profile in Table 1. If we take plurality as the voting rule, the minimal deciding set for candidate $a$ is $\{2{:}\{a,b\}, 3{:}\{b,c\}\}$. Firstly, it is a deciding set: if $\sigma(2{:}\{a,b\}) = a$ then $a$ is necessary winner; otherwise if $\sigma(2{:}\{a,b\}) = b$ and $\sigma(3{:}\{b,c\}) = c$, then $a$ is also a necessary winner; and otherwise if $\sigma(2{:}\{a,b\}) = b$ and $\sigma(3{:}\{b,c\}) = b$, then $a$ is a necessary loser.

Next we prove that all its proper subsets are not deciding sets. To prove this we only need to look at its subsets with size one. For $\{2{:}\{a,b\}\}$, a counterexample is when $\sigma(2{:}\{a,b\}) = b$. Given this answer, $a$ is both a possible winner and a possible loser in the new partial preference profile. Similarly for $\{3{:}\{b,c\}\}$, we get a counterexample when $\sigma(3{:}\{b,c\}) = b$.

Notice here that the comparison queries 2:$\{a,c\}$ and 3:$\{a,b\}$ are not in the minimal deciding set.

| | | | | | |
|---|---|---|---|---|---|
| 1 | $a$ | $>$ | $b$ | $>$ | $c$ |
| 2 | $b$ | $>$ | $c$ | | |
| 3 | $c$ | $>$ | $a$ | | |

Table 1: Partial preference profile

Sometimes one may also be interested in knowing the ways to make a candidate a winner or a loser in a vote. In this case, one may want to find sets of queries that when answered properly will lead to the candidate being a winner (or loser).

**Definition 5** *Let $p$ be a partial preference profile, $o$ a candidate, and $f$ a voting rule. A set $Q$ of queries is a possible winning (losing) set for $o$ (in $p$ under $f$) if there is an answer $\sigma$ such that $o$ is a necessary winner (loser) in the new partial profile $\sigma(p, Q)$ under $f$. $Q$ is a minimal possible winning (losing) set for $o$ if it is a possible winning (losing) set for $o$, and there is no other possible winning (losing) set $Q'$ for $a$ such that $Q' \subset Q$.*

For the example in Table 1, if we still use plurality as the voting rule, then $Q_1 = \{2{:}\{a,b\}\}$ is a possible winning set for $a$ to win because if we set $\sigma_1(2{:}\{a,b\}) = a$ then in the new partial profile $p(\sigma_1, Q_1)$ as shown in Table 2, $a$ is a necessary winner. Notice that it is not a deciding set. And this possible winning set is obviously minimal because its only proper subset $\emptyset$ is not a possible winning set for $a$.

The set $Q_2 = \{3{:}\{a,b\}\}$ is also a minimal possible winning set because when $\sigma_2(3{:}\{a,b\}) = a$ as shown in Table 3, then in the new partial profile $a$ is again a necessary winner. From this we can see that there could be multiple minimal possible winning sets for a candidate. Also notice that the query 3:$\{a,b\}$ is not in the minimal deciding set. So a minimal possible winning set may not have any overlap with the minimal deciding set.

| | | | | | |
|---|---|---|---|---|---|
| 1 | $a$ | $>$ | $b$ | $>$ | $c$ |
| 2 | $a$ | $>$ | $b$ | $>$ | $c$ |
| 3 | $c$ | $>$ | $a$ | | |

Table 2: $\sigma_1(p, Q_1)$

| | | | | | |
|---|---|---|---|---|---|
| 1 | $a$ | $>$ | $b$ | $>$ | $c$ |
| 2 | $b$ | $>$ | $c$ | | |
| 3 | $c$ | $>$ | $a$ | $>$ | $b$ |

Table 3: $\sigma_2(p, Q_2)$

It is easy to see that deciding sets always exist, and if $Q$ is a deciding set, and $Q \subseteq Q'$, then $Q'$ is also a deciding set. Furthermore, if $Q \neq \emptyset$, and $Q$ is a deciding set for $o$, then $Q$ is both a possible winning set and a possible losing set for $a$. But the converse is obviously not true in general.

In the following, we consider computing minimal deciding sets under the plurality and Borda rules. The case for the veto voting rule is similar to that of plurality.

# 4 Computing minimal deciding sets

If $Q$ is a deciding set for candidate $o$, then for any query $q$ not in $Q$, as far as the outcome for $o$ is concerned, the answer to $q$ is immaterial, thus can be totally ignored. This suggests that for any voting rule, any partial preference profile, and any candidate, there is a unique minimal deciding set for the candidate. This is indeed the case.

**Theorem 1** *For any voting rule $f$, partial preference profile $p$, and candidate $o$, there is a unique minimal deciding set for $o$ in $p$ under $f$.*

To prove this theorem, we need the following lemma about partial orders.

**Lemma 1** *Let $R$ be a partial order on $S$, and $a \neq b$ two elements in $S$ that are not comparable in $R$. Then there are two total orders $R_1$ and $R_2$ such that they both extend $R$, and are exactly the same except on $a$ and $b$: for any $x$ and $y$, $(x, y) \in R_1$ iff $(x, y) \in R_2$ provided $\{x, y\} \neq \{a, b\}$, and $(a, b) \in R_1$ but $(b, a) \in R_2$.*

**Proof of Theorem** Since the number of voters is finite, there exists a minimal deciding set $Q$ for $o$. Let $Q'$ be any other deciding set for $o$. If $Q$ is not a subset of $Q'$, then there is a $q \in Q$ but $q \notin Q'$. Let $Q_0 = Q \setminus \{q\}$. We show that $Q_0$ is also a deciding set. To show this, suppose $\sigma$ is an answer to $Q_0$ under $p$. We need to show that $o$ is either a necessary winner or a necessary loser in the new partial profile $p(\sigma, Q_0)$. Suppose $q$ is $i{:}\{x, y\}$ for some voter $i$ and candidates $x \neq y$. There are two cases:

1. The answer $\sigma$ already entails an answer to $q$, that is, either $(x, y)$ or $(y, x)$ is in $p_i(\sigma, Q_0)$. This basically means that $\sigma$ is also an answer to $Q$. Thus $o$ must be either a necessary winner or a necessary loser in the new partial profile $p(\sigma, Q_0)$ as $p(\sigma, Q_0) = p(\sigma, Q)$ and $Q$ is a deciding set.

2. Otherwise, by applying Lemma 1 to the partial order $p_i(\sigma, Q_0)$, we see that there are two answers $\sigma_1$ and $\sigma_2$ to $Q \cup Q'$ such that $\sigma_1$ and $\sigma_2$ are the same except on $q$ where we have $\sigma_1(q) = x$ and $\sigma_2(q) = y$. Since $q \notin Q'$, $\sigma_1$ and $\sigma_2$ are the same answer when restricted to $Q'$. Since $Q'$ is a deciding set, this means that $o$ must be either a necessary winner in $p(\sigma_1, Q')$ or a necessary loser in $p(\sigma_1, Q')$. Suppose $o$ is a necessary winner in $p(\sigma_1, Q')$. Then $o$ is also a necessary winner in $p(\sigma_2, Q')$ as $p(\sigma_1, Q')$ is the same as $p(\sigma_2, Q')$. It follows then that $o$ must also be a necessary winner in both $p(\sigma_1, Q \cup Q')$ and $p(\sigma_2, Q \cup Q')$. Since $Q$ is also a deciding set, $o$ is also a necessary winner in both $p(\sigma_1, Q)$ and $p(\sigma_2, Q)$. This means that $o$ is a necessary winner in $p(\sigma, Q_0)$. Similarly, if $a$ is a necessary loser in $p(\sigma_1, Q')$, then $o$ is also a necessary loser in $p(\sigma, Q_0)$.

From this theorem, we get the following corollary.

**Corollary 2** *If $Q_1$ and $Q_2$ are both deciding sets for a candidate $o$, then $Q_1 \cap Q_2$ is also a deciding set for $o$.*

Our next result provides a way to check if a query is in a minimal deciding set.

Suppose $S$ is the set of all comparison queries. Then trivially, $S$ is a deciding set for any candidate in any partial preference profile under any voting rule. Now consider any query $q \in S$, and any given candidate $o$ and partial profile $p$. Since there is a unique minimal deciding set for $o$ in $p$, it is clear that $q$ is in the minimal deciding set iff $S \setminus \{q\}$ is not a deciding set.

We thus have the following proposition.

**Proposition 1** *Let $S$ be the set of all (comparison) queries. For any candidate $o$, and any partial preference profile $p$, a query $q = i{:}\{a, b\}$ is in the minimal deciding set if and only if there is an answer $\sigma$ to $S \setminus \{q\}$ such that it can be extended to two answers $\sigma_1$ and $\sigma_2$ to $S$ such that $\sigma_1(q) = a$, $\sigma_2(q) = b$, and the outcome of $o$ is different in $p(\sigma_1, S)$ and $p(\sigma_2, S)$ (answer to the question "is $o$ is necessary winner or loser?" is different).*

This proposition will be used in our algorithm for computing the minimal deciding sets under the plurality rule.

## 4.1 Plurality

For the plurality and the veto rules, computing the minimal deciding set can be done in polynomial time. We show this for the plurality rule.

Based on Proposition 1, it suffices to check each query independently. Now, for a given query $q = i{:}\{o_1, o_2\}$, one may think that we need to check if there are two extensions $\sigma_1$ and $\sigma_2$ that are different only for $q$, with $a$ and $b$ ranked top by $i$ respectively, for every pair of candidates $a$ and $b$. Actually as plurality only concerns the number of times a candidate is ranked first, the answer to $q$ can affect the score vector only when $o_1$ and $o_2$ are ranked top in $i$'s vote in $\sigma_1$ and $\sigma_2$. Now we come down to a problem of whether there exists an extension of all votes except $i$'s such that when $i$'s vote is considered, the "outcome" for the targeted candidate $o$ changes (when $i$'s top choice is $o_1$ or $o_2$). This problem is not solely a flow problem because it concerns the score of two candidates. However, it can be reduced to a flow problem, as shown in EqualScore procedure below. Here is a detailed description of our algorithm.

Our algorithm makes use of an algorithm for MAX-FLOW problem introduced in [Cormen *et al.*, 2001]. The problems is, given a graph with capacity as numbers assigned to every edge, to determine the maximal amount of flow going from node $s$ to $t$ with the flow in each edge not exceeding its capacity. Here we use MAX-FLOW$(G, s, t)$ to denote the maximal flow from $s$ to $t$ in the flow graph $G$. Note that there are polynomial algorithms for MAX-FLOW$(G, s, t)$.

Given a partial preference profile $p$, we use $a >_i b$ to stand for $(a, b) \in p_i$. When we add some new preferences $a >_i b, c >_i d$, etc, to $p$, we mean that we get a new partial preference profile $p'$ such that $p'_j = p_j$ for every $j \neq i$, and $p'_i$ is the transitive closure of $p_i \cup \{(a, b), (c, d), \cdots\}$. When we delete some voters $i_1, i_2, \cdots, i_k$ from $p$, we mean that we get a new partial preference profile $p'$ such that the set of voters is $V \backslash \{i_1, i_2, \cdots, i_k\}$, and $p'_j = p_j$ for all $j \notin \{i_1, i_2, \cdots, i_k\}$. When we delete some candidates $o_1, o_2, \cdots, o_k$ from $p$, we mean that we get a new partial profile $p'$ such that the set of candidates is $O' = O \backslash \{o_1, o_2, \cdots, o_k\}$, and $p'_j$ is just $p_j$ constrained to $O'$.

**Algorithm**:QueryInMDS$(i{:}\{o_1, o_2\}, a, p)$

Input: a query $i{:}\{o_1, o_2\}$, a candidate $a$ and a partial preference profile $p$.
Output: *yes* or *no* of whether $i{:}\{o_1, o_2\}$ is in the minimal deciding set of $a$ in $p$.

1. If in $p$ there is a $w$ in $O \backslash \{o_1, o_2\}$ s.t. $w >_i o_1$ or $w >_i o_2$, then return *no*.
2. Else if $a \notin \{o_1, o_2\}$, then we do the following. First, let $p^1$ be the profile we get by adding $o_1 >_i o_2$ and $o_2 >_i c$ for every $c \notin \{o_1, o_2\}$ to $p$. If EqualScore$(a, o_2, p^1) =$ *yes*, then return *yes*, else let $p^2$ be the profile we get by adding $o_2 >_i o_1$ and $o_1 >_i c$ for every $c \neq o_1, o_2$ into $p$. If EqualScore$(a, o_1, p^2) =$ *yes*, then return *yes*, else return *no*.

3. Else, $a \in \{o_1, o_2\}$. W.L.O.G, let $a = o_1$. The case for $a = o_2$ is exactly the same. Let $p^3$ be the profile we get by adding $a >_i c$ for all alternative $c \neq a$ to $p$. If EqualScore$(a, m, p^3) =$ *yes* for some candidate $m \in O, m \neq a$, then return *yes*, else let $p^4$ be the profile we get by deleting voter $i$ from $p$. $p^4$ has one less voter than $p$. If EqualScore$(a, o_2, p^4) =$ *yes*, then return *yes*, else return *no*.

**Algorithm**:EqualScore$(a, b, p)$

Input: candidates $a$ and $b$ and a partial preference profile $p$.
Output: *yes* or *no* of whether there is a completion $p'$ of $p$ s.t. the scores of $a$ and $b$ in $p'$ are the same and the maximal among all candidates.

Let $S_a = \{i \mid \neg\exists w \in O\backslash\{a\}, w >_i a \in p\}$, $S_b = \{i \mid \neg\exists w \in O\backslash\{b\}, w >_i b \in p\}$, $S_i = S_a \cap S_b$, $s_a = |S_a|$, $s_b = |S_b|$, $s_i = |S_i|$.

1. If $|s_a - s_b| \leq s_i$ and $|s_a + s_b - s_i| \bmod 2 = 0$, then let $p'$ be the new profile we get by deleting all the voters in $S_a \cup S_b$ and candidates $a$ and $b$ from $p$, and $T = |s_a + s_b - s_i|/2$. If Graph$(p', T) =$ *yes*, return *yes*, else return *no*.

2. Else if $|s_a - s_b| \leq s_i$ and $|s_a + s_b - s_i| \bmod 2 \neq 0$, then let $T = (|s_a + s_b - s_i| - 1)/2$. For every $i \in S_a \cup S_b$, let $p'$ be the profile we get from deleting all the voters in $S_a \cup S_b \backslash \{i\}$ and candidates $a$ and $b$ from $p$. If Graph$(p', T) =$ *yes* then return *yes*. If none of these return *yes*, then return *no*.

3. Else we have $|s_a - s_b| > s_i$. W.L.O.G., let $s_b > s_a$. The case for $s_a > s_b$ is exactly the same. Let $p'$ be the profile obtained by deleting all the votes in $S_a$ and candidate $a$ from $p$ and set $T = s_a$. If Graph$(p', T) =$ *yes*, then return *yes*, else return *no*.

**Algorithm**:Graph$(p, T)$.

Input: a partial preference profile $p$ and a threshold $T$.
Output: *yes* or *no* of to indicate whether there is a completion $p'$ of $p$, in which the maximal score of all candidates in $p'$ is $\leq T$.

Let $N$ be the set of voters and $O$ the set of candidates in $p$. Let $s$ and $t$ be two new atoms not in $N \cup O$. Construct a flow graph $G$ with $\{s, t\} \cup N \cup O$ as the set of nodes, and the following three layers of edges:

1. For every node in $N$, an edge from $s$ to it with capacity one.

2. For every node $i \in N$ and every node $o \in O$ s.t. $\neg\exists o' \in O, o' >_i o$, an edge from $i$ to $o$ with capacity one.

3. For every $o \in O$, an edge from it to $t$ with capacity $T$.

If MAX-FLOW$(G, s, t) = |N|$, then return *yes*, else return *no*.

**Lemma 2** *Graph$(p, T)$ returns yes iff $p$ has a completion with every candidate getting at most score $T$ under plurality.*

**Lemma 3** *EqualScore$(a, b, p)$ returns yes iff there is a completion $p_c$ of $p$ s.t. the scores of $a$ and $b$ in $p_c$ are both the maximal score under plurality.*

We omit the proofs of these two lemmas here because of the page limit.

**Corollary 3** *QueryInMDS$(i{:}\{o_1, o_2\}, a, p)$ returns yes iff $i{:}\{o_1, o_2\}$ is in the minimal deciding set of $a$ in $p$ under plurality, and it runs in polynomial time.*

The number of edges in the graph in procedure Graph is $O(mn)$, and the max flow found by Graph is $O(n)$. So if we use FORD-FULKERSON algorithm in [Cormen *et al.*, 2001] to implement MAX-FLOW, Graph runs in $O(mn^2)$ time. EqualScore calls Graph for at most $n$ times, so the complexity of EqualScore is $O(mn^3)$. QueryInMDS calls EqualScore for $O(m)$ times, so QueryInMDS runs in $O(m^2n^3)$ time.

As plurality only concerns the candidate ranked first by every voter, according to Proposition 1, $i$:$\{o_1, o_2\}$ is in the minimal deciding set of $a$ iff there are two assignments of a top choice for every voter, $\tau_1$ and $\tau_2$, which are consistent with $p$ s.t. $\{\tau_1(i), \tau_2(i)\} = \{o_1, o_2\}$, and $\forall j \neq i, \tau_1(j) = \tau_2(j)$ and $a$ is winner under assignment $\tau_1$ but loser under $\tau_2$. Firstly, we prove the "$\Rightarrow$" part of the corollary. In step 1, if the procedure does not return no then $o_1$ and $o_2$ are both legal top choices for $i$ in $p$. In step 2, if the algorithm returns yes, then w.l.o.g we have EqualScore$(a, o_2, p^1)$ = yes so there is an evidence $\tau_1$ assigning the maximal number of votes to both $a$ and $o_2$ with $o_1$ the top choice of voter $i$. This is just the evidence $\tau_1$ in which $a$ is a winner. And we can change $\tau_1(i)$ into $o_2$ to get $\tau_2$, in which $a$ is a loser. Notice that only $\tau_2$ and $\tau_1$ are different only on $i$. So $i$:$\{o_1, o_2\}$ is in the minimal deciding set of $a$. In step 3, the algorithm returns yes in two cases: EqualScore$(a, m, p^3)$ is true for some $m \in O$ or EqualScore$(a, o_2, p^4)$ is true. Here w.l.o.g we suppose $a = o_1$. If EqualScore$(a, m, p^3)$ is true, then we have an assignment $\tau_1$ in which $a$ and $m$ both have the maximal score among all candidates and $\tau_1(i) = a$. We can just change voter $i$'s top choice from $a$ into $o_2$ to get $\tau_2$. And $a$ is winner under $\tau_1$ but loser under $\tau_2$. So, again $i$:$\{o_1, o_2\}$ is in the minimal deciding set of $a$ in $p$. If EqualScore$(a, o_2, p^4)$ returns yes, then we have an assignment of candidates to every voter except $i$, such that $a$ and $o_2$ have equal maximal score. This is an incomplete assignment of top choices $\tau'$. Combining $\tau'$ with $\tau_1(i) = a$ we get $\tau_1$ in which $a$ wins, while combining it with $\tau_2(i) = o_2$ we get $\tau_2$ in which $a$ loses. So we can conclude $i$:$\{o_1, o_2\}$ is in the minimal deciding set of $a$.

Then we prove the "$\Leftarrow$" part of the corollary. As we just argued in the previous paragraph, there are two assignments of top choice $\tau_1$ and $\tau_2$ for every voter as we described. Suppose we record score of a candidate $c$ under $\tau_1$ and $\tau_2$ as $s_c^1$ and $s_c^2$, and the maximal score among all candidates under $\tau_1$ and $\tau_2$ as $s_{\max}^1$ and $s_{\max}^2$. Notice that we always have $|s_a^1 - s_a^2| \leq 1, |s_{\max}^1 - s_{\max}^2| \leq 1$ and $s_a^1 = s_{\max}^1$ and $s_a^2 < s_{\max}^2$. If $a \notin \{o_1, o_2\}$, then $s_a^1 = s_a^2$, so $s_{\max}^2 - s_{\max}^1 = 1$, and $s_{\max}^1 = s_a^1$. As the $s_{\max}^2 > s_{\max}^1$ and w.l.o.g only $o_2$ has a higher score in $\tau_2$ than in $\tau_1$, so we have that $s_{o_2}^1 = s_{\max}^1 = s_a^1$ and $s_{o_2}^2 = s_{\max}^2$. Notice that under $\tau_1$ which is consistent with $p$, $o_2$ and $a$ both have maximal score among all candidates, so EqualScore$(a, o_2, p^2)$ = yes and so our algorithm will return yes in step 2.

If $a \in \{o_1, o_2\}$, then by a similar analysis we could conclude that our algorithm will also return yes. So we have proven Corollary 3. As there are only polynomial such queries, computing the minimal deciding set is also in P.

If $a \in \{o_1, o_2\}$, then w.l.o.g, we suppose $a = o_1$. Similarly, $s_a^1 = s_a^2 + 1$ and $s_{o_2}^2 = s_{o_2}^1 + 1$. Also, $|s_{\max}^1 - s_{\max}^2| \leq 1$ and $s_a^1 = s_{\max}^1$ and $s_a^2 < s_{\max}^2$. If $a$ has unique maximal score in $\tau_1$, then we have $s_{o_2}^1 = s_a^1 - 1$ because only $o_2$ has a higher score under $\tau_2$ than under $\tau_1$. So our algorithm will return yes because EqualScore$(a, o_2, p^4)$ = yes. If some other candidate $m$ also has maximal score under $\tau_1$, then it will also return yes because EqualScore$(a, m, p^3)$ = yes for candidate $m$.

## 4.2 Borda and other scoring voting rules

On the other hand, under the Borda voting rule and other scoring rules, computing minimal deciding sets is NP-complete.

The Borda voting rule uses the score vector $W = \{m, m-1, \cdots, 1\}$, where $m$ is the number of candidates. Given a partial preference profile $p$, and a candidate $o$, it is known that checking if $o$ is a possible winner in $p$ under Borda is an NP-complete problem [Xia and Conitzer, 2011]. It came as no surprise that checking whether a query $q$ is in the minimal deciding set is also an NP-complete problem.

**Theorem 4** *The problem of checking if a query $q$ is in the minimal deciding set for a candidate $o$ in a partial profile $p$ under the Borda voting rule is NP-complete.*

**Proof** The problem is in NP follows from Proposition 1 as checking whether an answer $\sigma$ to $S \setminus \{q\}$ is legal in $p$, whether it can be extended to two different answers to $q$ such that the outcomes of $o$ in the two extensions are different under the Borda rule can all be done in polynomial time, where $S$ is the set of all queries.

The problem is NP-hard because $o$ is a possible winner iff either $o$ is a necessary winner or the minimal deciding set for $o$ is not empty. Notice that checking if $o$ is a necessary winner under Borda can be done in polynomial time [Konczak and Lang, 2005].

In fact, the proof of this theorem gives a more general result:

**Theorem 5** *For any polynomial time voting rule under which the possible winner problem is NP-complete and the necessary winner problem is in P, the problem of checking if a query is in the minimal deciding set is NP-complete.*

It is known that except for plurality and veto rules, all scoring rules have the property in the above theorem [Xia and Conitzer, 2011; Betzler and Britta, 2010; Baumeister and Rothe, 2010]. We thus have the following corollary.

**Corollary 6** *For any scoring voting rule that is different from the plurality and the veto rules, checking if a query is in the minimal deciding set is NP-complete.*

# 5 Computing possible winning sets

We have seen that there may be multiple minimal possible winning sets. This makes the problem of computing these sets harder.

**Proposition 2** *A candidate $o$ is a possible winner in a partial preference profile $p$ iff the set of all queries is a possible winning set for $a$. A candidate $o$ is a possible loser iff she is not a necessary winner iff the set of all queries is a possible losing set for $a$.*

Thus just like Corollary 6, we have the following result.

**Theorem 7** *For any scoring voting rule that is different from the plurality and the veto rules, checking if a set of queries is a possible winning set is NP-complete.*

We do not at present know the complexity of computing a minimal possible winning set. Our guess is that it is $\Pi_2^P$-complete, same as the complexity of computing minimal models (circumscription) in propositional logic [Eiter and Gottlob, 1993].

Neither do we know the exact complexity of checking if a set of queries is a possible losing set for a candidate. While Proposition 2 implies that checking if the set of all queries is a possible losing set for a candidate is in P, the problem seems to be harder in general as it requires checking whether an answer has enough information to conclude that the candidate is a necessary loser, which is a coNP-complete problem for voting rules such as Borda.

We now show that for the plurality voting rule, deciding whether a query set is a minimal possible winning set is in P.

### 5.1 Plurality

**Algorithm**: PossibleWinningSet$(Q, p, a)$:

Input: A query set $Q$, a partial profile $p$ and a candidate $a$, and we assume that $\forall i$:$\{b, c\} \in Q, (b, c) \notin p_i, (c, b) \notin p_i$.
Output: *yes* or *no* of whether $Q$ is a possible winning set of $a$ in $p$.

1. For every voter $i$, let $G_i$ be the undirected graph with $O$ as nodes and $\{(b, c) \mid i$:$\{b, c\} \in Q\}$ as edges and $S_i$ be the set of all strongly connected components of $G_i$. For a strongly connected component $u$, we use $V(u)$ to represent the set of vertices of $u$. For every voter $i$, for every strongly connected component $u \in S_i$ s.t. $a \in V(u)$ and $\neg\exists o \in O, o >_i a$, add $a >_i c$ to $p$ for every $c \neq a$ in $V(u)$. Set $s_a =$ the minimal score of $a$ in $p$.

2. $\forall i \in N, U_i = \{u \in S_i \mid \forall o \in V(u), \neg\exists w \in O, w >_i o\}, U = U_1 \cup \cdots \cup U_n$

3. Let $O$ be the set of candidates in $p$ and $U$ as defined. Let $s$ and $t$ be two new atoms not in $U \cup O$. Construct a graph $G$ with $\{s, t\} \cup U \cup O \backslash \{a\}$ as set of nodes, and the following three layers of edges:

   (a) for every node in $U$ an edge from $s$ to it with capacity one.

   (b) for every node $u \in U$ and every candidate $o$ s.t. $o \in O \backslash \{a\}$ and $o \in V(u)$, an edge from $u$ to $o$ with capacity one.

   (c) for every $o \in O \backslash \{a\}$ an edge from it to $t$ with capacity $s_a$.

   If MAX-FLOW$(G, s, t) = |U|$, then return *yes*, else return *no*.

**Corollary 8** *PossibleWinningSet*$(Q, p, a)$ *returns yes iff* $Q$ *is a possible winning set of* $a$ *in* $p$*, and it runs in polynomial time.*

As proven in [Konczak and Lang, 2005], $a$ is a necessary winner in $p$ *iff* the minimal score of $a$ is higher than the maximal score of any other candidate $c \in O$.

Intuitively, our algorithm tries to maximize the *min score* of $a$ and see whether the *max score* of other candidates can be less than the min score of $a$ under some answer of $Q$. The detailed proof of the correctness of the algorithm is omitted due to page limit.

The graph constructed has $O(mn)$ edges, and the flow found by the algorithm is $O(mn)$. So if we use FORD-FULKERSON alogrithm to implement MAX-FLOW, the flow calculation takes $O(m^2n^2)$ time. And the strongly connected components part runs in $O(m^2n)$ time as SCC is $O(|V| + |E|)$ and the size of the graph is $O(m^2)$. So PossibleWinningSet runs in $O(m^2n^2)$ time.

To determine whether $Q$ is a minimal possible winning set, we just need $|Q|+1$ calls of the above algorithm. So determining whether a query set is a minimal possible winning set under plurality is also in P.

## 6 Related works

We have mentioned that this work generalizes the notions of necessary and possible winners [Konczak and Lang, 2005]. It is also closely related to work on vote elicitation (e.g. [Conitzer and Sandholm, 2002; Procaccia, 2008]). In vote elicitation, one is often interested in a dynamic question and answering process [Conitzer and Sandholm, 2002]. Here we are looking at statically, in the current state, how many possible questions one needs to ask in order to determine the outcome of a vote w.r.t. to a particular candidate. These two approaches are closely related. For instance, given our notion of minimal deciding sets, we can proceed in the following way to decide the outcome for a candidate $x$: in the current state, find a query that is in the minimal deciding set of $x$, ask the query and add the answer to the current partial preference profile; repeat this in the new state until one reaches a state where $x$ is either a necessary winner or a necessary loser. The dynamic process thus obtained seems to be new, and we plan to explore its properties and connections with existing dynamic approaches in our future work.

Following the theoretical study of vote elicitation, researchers are recently doing some experimental studies of elicitation processes (e.g. [Lu and Boutilier, 2011a; 2011b; Kalech *et al.*, 2011]). In the these work, the focus is mainly to save the number of questions in and rounds of the elicitation process, and to develop approximations when the partial information is not enough to decide the winner. In contrast to these approaches, ours may be more easily parallelised and more efficient when we only care about one candidate.

# 7 Concluding remarks

We have considered sets of questions to ask the voters about in order to determine the outcome of a vote with partial information.

A deciding set is one that will determine the outcome of a vote for a candidate no matter how the queries in the set are answered. One fundamental property about this notion is that among these sets, there is a unique minimal one. Thus as far as a candidate is concerned, a comparison between two candidates is irrelevant to her if the associated query is not in her minimal deciding set. Computationally we have shown that the minimal deciding set can be computed in polynomial time for the plurality and veto rules, and is NP-complete to compute for other scoring rules. We will study complexity for other voting rules in our future work.

On the other hand, a possible winning (losing) set for a candidate is one that has an answer that will lead to the candidate being a necessary winner (loser). For a manipulator, these sets may be of more interest as they could tell her how to influence the voters to make the candidate a winner or a loser of the vote. We have shown that for plurality and veto rules, a minimal possible winning set can be computed in polynomial time. We believe that the same is true for computing a minimal possible losing set as well. For scoring voting rules such as Borda, the problem is again NP-hard for checking if a set of queries is a possible winning set.

# References

[Baumeister and Rothe, 2010] Dorothea Baumeister and Jörg Rothe. Taking the final step to a full dichotomy of the possible winner problem in pure scoring rules. In *ECAI-10*, pages 1019–1020, Amsterdam, The Netherlands, The Netherlands, 2010. IOS Press.

[Betzler and Britta, 2010] Nadja Betzler and Dorn Britta. Towards a dichotomy for the possible winner problem in elections based on scording rule. *Journal of Computer and System Sciences*, 76:812–836, 2010.

[Boutilier *et al.*, 2004] Craig Boutilier, Ronen I. Brafman, Carmel Domshlak, Holger H. Hoos, and David Poole. Cp-nets: A tool for representing and reasoning with conditional *Ceteris Paribus* preference statements. *JAIR*, 21:135–191, 2004.

[Chevaleyre *et al.*, 2007] Yann Chevaleyre, Ulle Endriss, Jrme Lang, and Nicolas Maudet. A short introduction to computational social choice. In Jan van Leeuwen, Giuseppe Italiano, Wiebe van der Hoek, Christoph Meinel, Harald Sack, and Frantiek Plil, editors, *SOFSEM 2007: Theory and Practice of Computer Science*, volume 4362 of *Lecture Notes in Computer Science*, pages 51–69. Springer Berlin / Heidelberg, 2007.

[Conitzer and Sandholm, 2002] Vincent Conitzer and Tuomas Sandholm. Vote elicitation: Complexity and strategy-proofness. In *AAAI-02*, pages 392–397. AAAI, 2002.

[Conitzer *et al.*, 2011] Vincent Conitzer, Toby Walsh, and Lirong Xia. Dominating manipulations in voting with partial information. In *IJCAI-2011 Workshop on Social Choice and Artificial Intelligence*, 2011.

[Conitzer, 2009] Vincent Conitzer. Eliciting single-peaked preferences using comparison queries, 2009.

[Cormen *et al.*, 2001] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Second Edition*, chapter 26. MIT Press, 2001.

[Eiter and Gottlob, 1993] Thomas Eiter and Georg Gottlob. Propositional circumscription and extended closed world reasoning are $\Pi_2^p$-complete. *Theoretical Computer Science*, 114:231–245, 1993.

[Kalech *et al.*, 2011] Meir Kalech, Sarit Kraus, Gal A. Kaminka, and Claudia V. Goldman. Practical voting rules with partial information. *Autonomous Agents and Multi-Agent Systems*, 22(1):151–182, January 2011.

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

[Lu and Boutilier, 2011a] Tyler Lu and Craig Boutilier. Robust approximation and incremental elicitation in voting protocols. In *IJCAI-11*, pages 287–293, 2011.

[Lu and Boutilier, 2011b] Tyler Lu and Craig Boutilier. Vote elicitation with probabilistic preference models: empirical estimation and cost tradeoffs. In *Proceedings of the Second international conference on Algorithmic decision theory*, ADT'11, pages 135–149, Berlin, Heidelberg, 2011. Springer-Verlag.

[Pini *et al.*, 2007] M.S. Pini, F. Rossi, K. B. Venable, and T. Walsh. Incompleteness and incomparability in preference aggregation. IJCAI-07, pages 1464–1469, 2007.

[Procaccia, 2008] Ariel D. Procaccia. *Computational Voting Theory: Of the Agents, By the Agents, For the Agents.* PhD thesis, The Hebrew University of Jerusalem, Sep. 2008.

[Xia and Conitzer, 2011] Lirong Xia and Vincent Conitzer. Determining possible and necessary winners under common voting rules given partial orders. *JAIR*, 41:25–67, May 2011.

Ning Ding
Department of Computer Science and Engineering
The Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong
Email: dning@cse.ust.hk

Fangzhen Lin
Department of Computer Science and Engineering
The Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong
Email: flin@cse.ust.hk