Vote Elicitation: Complexity and Strategy-Proofness*

Vincent Conitzer Tuomas Sandholm
Computer Science Department
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213
{conitzer, sandholm}@cs.cmu.edu

June 12, 2002

Abstract

Preference elicitation is a central problem in AI, and has received significant attention in single-agent settings. It is also a key problem in multiagent systems, but has received little attention here so far. In this setting, the agents may have different preferences that often must be aggregated using voting. This leads to interesting issues because what, if any, information should be elicited from an agent depends on what other agents have revealed about their preferences so far. In this paper we study effective elicitation, and its impediments, for the most common voting protocols. We analyze the complexity of determining whether enough information has been elicited, as well as the complexity of deciding whose votes to elicit. We study these questions for unweighted and weighted voters, with uniform and heterogeneous elicitation costs across voters, and with a constant as well as with an unbounded number of candidates in the election. We also show

*This material is based upon work supported by the National Science Foundation under CAREER Award IRI-9703122, Grant IIS-9800994, ITR IIS-0081246, and ITR IIS-0121678. A short early version of this paper has been accepted for publication at the National Conference on Artificial Intelligence (AAAI), Edmonton, Canada, 2002.

that elicitation introduces additional opportunities for strategic manipulation by the voters. We demonstrate how to curtail the space of elicitation schemes so that no such additional strategic issues arise.

1 Introduction

Preference elicitation is a central problem in AI. To build a bot that acts intelligently on behalf of any type of agent (a human, a corporation, a software agent, etc.), the bot needs to know about the agent's preferences. However, the bot should only elicit pertinent preference information from the agent because determining and expressing preferences can be arduous. Significant work has been done on selective preference elicitation (e.g., [2, 3, 8–10, 17, 23, 24, 28, 29]).

Preference elicitation is also a key problem in multiagent systems, but has received little attention so far.[1] The agents may have different preferences over the set of *candidates* that the agents must collectively choose among (e.g., potential presidents, joint plans, resource allocations, task allocations, etc.). The most general method for aggregating preferences is voting.[2] In traditional voting, each voter is asked for its complete preferences. We observe that intelligently eliciting preferences from the voters can allow the voting protocol to determine the outcome well before all of the preferences have been elicited. This is desirable for any of several reasons: 1) it can be costly for an agent to determine its own preferences (e.g., computationally [18, 20, 25]), 2) communicating the preferences introduces overhead (network traffic, traveling to the voting booth to vote, traveling door to door to collect votes, etc.), and 3) less preference revelation is desirable due to privacy reasons.

Attempting to efficiently elicit preferences leads to interesting issues in the voting context because what, if any, information should be elicited from an agent depends on what other agents have revealed about their preferences so far. The goal here is to determine the right outcome while eliciting a minimal amount of preference information from the voters. The most effective elicitation schemes make use of *suspicions* about the agents' preferences.

[1] A notable exception is bid elicitation in combinatorial auctions (e.g., [4–6, 16]) and exchanges [27]. Market mechanisms where agents reveal their demand (or price) on goods at every iteration based on price feedback (or quantity feedback) can also be viewed as a form of incremental preference revelation (e.g., [21, 22, 25, 26, 30, 31]).

[2] Voting mechanisms have been used also for computational agents (e.g., [11–15]).

Such suspicions can be the result of votes in previous elections, an understanding of the candidates in the election, an understanding of how each agent relates to each candidate, etc. To see how suspicions may help the elicitation process, consider a simple election with two candidates. If the elicitor knew beforehand which agents would vote for the eventual winner, simply querying enough of those voters would suffice.

In this paper we analyze the possibility of effective vote elicitation, and demonstrate two categories of impediments. First, optimal elicitation can be computationally complex. In Section 4 we show that even determining whether enough information has been elicited is $\mathcal{N}P$-complete for some voting protocols. In Section 5 we show that for most of the voting protocols, determining an efficient elicitation policy is $\mathcal{N}P$-complete (even with perfect suspicions). Second, in Section 6 we show that in various ways, elicitation can introduce additional opportunities for strategically manipulating the election. We then show how to avoid such problems by curtailing the space of elicitation schemes.

2 Common voting protocols

In this section we summarize the common voting protocols that we analyze. We consider elections with m candidates and n voters (agents). A voting protocol defines a function from the set of all possible combinations of votes to the set of candidates, the *winner determination function*. We now review the most common protocols in use, all of which will be studied in this paper.

- *Plurality.* Each candidate receives one point for each voter that ranked it first. The candidate with the most points wins.

- *Borda.* For each voter, a candidate receives $m-1$ points if it is the voter's top choice, $m-2$ if it is the second choice, $\ldots$, 0 if it is the last. The candidate with the most points wins.

- *Copeland (aka. Tournament).* The protocol simulates a pairwise election for each pair of candidates in turn (in a pairwise election, a candidate wins if it is preferred over the other candidate by more than half of the voters). A candidate gets 1 point if it defeats an opponent, 0 points if it draws, and -1 points if it loses. The candidate with the most points wins.

- *Maximin.* A candidate's *score* in a pairwise election is the number of voters that prefer it over the opponent. A candidate's number of points is its lowest score in any pairwise election. The candidate with the most points wins.
- *Single Transferable Vote (STV).* The winner determination process proceeds in rounds. In each round, a candidate's score is the number of voters that rank it highest among the remaining candidates, and the candidate with the lowest score drops out. The last remaining candidate wins. (The name comes from the fact that a vote *transfers* from its top remaining candidate to the next highest remaining candidate when the former drops out.)
- *Approval.* Each voter labels each candidate as either approved or disapproved. The candidate that is approved by the largest number of voters wins.

3 Definition of elicitation

In this section, we formally define elicitation. We distinguish between *full elicitation*, where the entire vote is elicited from every agent; *coarse elicitation*, where upon querying an agent the elicitor always asks for the agent's entire vote; and *fine elicitation* where this need not be the case (for example, an agent may be asked only what its most preferred candidate is). We formalize elicitation policies as trees.

Definition 1 *A* coarse elicitation tree *is a tree with the following properties:*

- *Each nonleaf node v is labeled with an agent a_v to be queried.*
- *Each nonleaf node v has a child for each of the possible votes of agent a_v.*
- *On each path from the root to a leaf, each agent occurs at most once.*

This tree determines how the elicitation will proceed for any combination of votes by the agents. The elicitor starts at the root. At each node, it queries the corresponding agent, and subsequently moves to the child corresponding to the obtained vote. We say the tree is *valid* for a protocol when for each leaf, given the votes corresponding to the path from the root to that leaf, the election's outcome is determined.

Definition 2 *A* fine elicitation tree *is a tree with the following properties:*

- *Each nonleaf node v is labeled with an agent a_v to be queried, a subset of that agent's possible votes S_v (the ones consistent with a_v's responses so far), and a query to be asked at that node. The query is given by a partition $\mathcal{T}_v$ of S_v; once the query is answered, one element of $\mathcal{T}_v$ is the set of remaining consistent votes.*

- *Each nonleaf node v has a child for each element of $\mathcal{T}_v$.*

- *Given a nonleaf v, if a_v does not occur anywhere else on the path from the root to v, then S_v is the set of all possible votes by a_v; otherwise, consider the node w closest to v on that path with $a_v = a_w$. The element of $\mathcal{T}_w$ corresponding to w's child on the way to v must equal S_v.*

- *Each partition $\mathcal{T}_x$ has at least 2 elements.*

The interpretation is as follows. Each node still corresponds to a query to the corresponding agent. A subset at a node is the set of the agent's possible votes that are consistent with its responses so far. The partition indicates the various ways in which this set may be reduced through the query.

We say the tree is *valid* for a protocol if for each leaf, the outcome of the election is determined by the responses to the queries on the path to that leaf. From now on, we only consider valid trees.

Our model of elicitation is very general. It can be used to represent intuitively reasonable queries as well as baroque ones such as *"Is it true that a is your most preferred candidate* or *that you prefer b to c?"* (which could impose a computational burden on the voter disproportionate to the fact that it is only one query). Reasonable fine elicitation policies will have some restriction on the types of $\mathcal{T}_v$ allowed. Also, elicitation trees can be extremely large. Therefore, it can be unreasonable to expect the elicitor to use this explicit representation for its elicitation policy, much less to do an exhaustive search over these trees to find one that minimizes the number of queries (for example, in the average case). Nevertheless, each well-defined elicitation policy corresponds to an elicitation tree, and hence elicitation trees are useful tools for analysis.

4 Hardness of terminating elicitation

Any sensible elicitation policy would need to be able to determine when it can safely terminate in the sense that the winning candidate can be determined from the elicited votes. In this section, we first show that for the STV protocol, it can be hard to determine when the elicitation process can terminate. We show this for unweighted voters and an unbounded number of candidates in Section 4.1, and for weighted voters and a constant number of candidates in Section 4.2. We then show that terminating elicitation is easy for each of the other voting protocols under study (Section 4.3).

4.1 Hardness of terminating elicitation in STV with equal weights

First we need a formal definition of the elicitation termination problem.

Definition 3 (ELICITATION-NOT-DONE) *We are given a set of votes S, a number t of votes that are still unknown, and a candidate h. We are asked whether there is a way to cast the t votes so that h will not win.*

In order to prove our hardness result, we make use of the following result from the literature on the difficulty of manipulating an election.

Definition 4 (EFFECTIVE-PREFERENCE) *We are given a set of votes S and a candidate c. One vote is not yet known. Is there a way to cast the last vote that makes c win?*

Theorem 1 (Known [1]) *For the STV protocol, EFFECTIVE-PREFERENCE is $\mathcal{NP}$-complete, even under the restriction that at least one of the votes in S puts c in the top spot.*

The restriction is of little interest in itself, but we will use it for our reduction.

Theorem 2 *For the STV protocol, ELICITATION-NOT-DONE is $\mathcal{NP}$-complete, even when $t = 1$.*[3]

[3] In all $\mathcal{NP}$-completeness proofs, we only prove $\mathcal{NP}$-hardness because proving that the problem is in $\mathcal{NP}$ is trivial.

Proof: It is easy to show that the problem is in $\mathcal{N}P$. To show $\mathcal{N}P$-hardness, we reduce an arbitrary instance of EFFECTIVE-PREFERENCE (with the restriction that at least one of the votes in S puts c at the top) to an instance of ELICITATION-NOT-DONE as follows. In the EFFECTIVE-PREFERENCE instance, let the candidate set be C_{EP} and the set of given votes S_{EP}. Then, in our ELICITATION-NOT-DONE instance, the candidate set is $C_{EP} \cup \{h\}$. The known (elicited) set of votes S includes all the votes from S_{EP}, where h is appended to these votes at the bottom – with the exception that one of the votes with c at the top inserts h into the second place (right behind c). Additionally, S includes $|S_{EP}|$ additional votes which place h in the top spot and rank the other candidates in whichever order. Finally, we set $t = 1$. We prove the instances are equivalent by making the following observations. First, h will always survive until the last round as it has almost half the votes at the start. Second, if there exists a way for the last vote to be cast such that h does not win the election, we may assume that this vote places h at the bottom, since if this vote ever transferred to h, h would win the election as it would hold more than half the votes. Third, h will not win the election if and only if it faces c in the last round (if c gets eliminated, the vote that ranks h right below c would transfer to h and h would win the election; on the other hand, c is ranked above h in all the votes that do not put h at the top, so c would win the last round). Fourth, as long as c remains in the election, the score of each candidate (besides h) in each round before the last will be exactly the same as the corresponding score in the EFFECTIVE-PREFERENCE instance (if we give the same value to the unknown vote in both instances). This follows from the fact that in this case, no vote will ever transfer to or from h and the relevant votes are identical otherwise. It follows that the remaining vote can be cast in such a way as to lead c to the final round if and only if the remaining vote in the EFFECTIVE-PREFERENCE instance can make c win the election. But then, by our third observation, the instances are equivalent. ■

4.2 Hardness of terminating elicitation in STV with a constant number of candidates and unequal weights

In this subsection, we will show that if voters are weighted, it can be hard to decide whether we can terminate elicitation even when the number of candidates is 4. We first need to define the weighted version of the problem.

Definition 5 (WEIGHTED-ELICITATION-NOT-DONE) *We are given a set of known votes S with weights w_i, a set T of voters with weights w_i whose votes are still unknown, and a candidate h. We are asked whether there is a way to cast the votes in T so that h will not win.*

Theorem 3 *For the STV protocol, WEIGHTED-ELICITATION-NOT-DONE is $\mathcal{NP}$-complete, even when the number of candidates is* 4.

Proof: Consider the DESTRUCTIVE-COALITIONAL-WEIGHTED-MANIPULATION problem as defined in [7]. In that problem, we are given a set of known votes S with weights w_i (corresponding to the nonmanipulating voters), a set T of voters with weights w_i (corresponding to the manipulating voters), and a candidate h. We are asked whether the manipulators can cast their votes so that h does not win. It has been shown that DESTRUCTIVE-COALITIONAL-WEIGHTED-MANIPULATION is $\mathcal{NP}$-complete, even when the number of candidates is 4 [7].

We observe that elicitation is done if and only if the unknown voters cannot "destructively manipulate" the election. This establishes the equivalence between the problems WEIGHTED-ELICITATION-NOT-DONE and DESTRUCTIVE-COALITIONAL-WEIGHTED-MANIPULATION. Therefore, WEIGHTED-ELICITATION-NOT-DONE is $\mathcal{NP}$-complete, even when the number of candidates is 4. ∎

Theorems 2 and 3 apply to both fine and coarse elicitation because in both the elicitor might end up in a situation where it has elicited some votes completely and others not at all.

With unweighted voters and a constant number of candidates, terminating STV is easy (polynomial time). Because votes are interchangeable, the elicitor can simply enumerate all effectively different vote combinations for the remaining t votes, and check each one. There are $\binom{m!+t-1}{m!-1} \leq (t+1)^{m!}$ of them (a standard combinatorial identity for the number of ways t indistinguishable balls can be placed into $m!$ distinguishable bins), which is exponential only in m, hence polynomial for constant m.

4.3 Ease of terminating elicitation in other voting protocols

For all the other voting protocols discussed in this paper, it can be determined in polynomial time whether elicitation can be terminated, even when the votes have different weights.

Theorem 4 *Consider any voting protocol where each candidate receives a numerical score based on the votes, and the candidate with the highest score wins. Suppose that the score function is monotone, that is, if voter i changes its vote so that $\{b : a >_i^{old} b\} \subseteq \{b : a >_i^{new} b\}$, a's score will not decrease. Finally, assume that the winner can be determined in polynomial time. Then for this protocol, WEIGHTED-ELICITATION-NOT-DONE can be solved in polynomial time, by checking what the outcome would be for $O(m)$ possible block votes by the yet unelicited voters. (In a block vote, everyone votes the same way.)*

Proof: The analogous theorem has already been proven for DESTRUCTIVE-COALITIONAL-WEIGHTED-MANIPULATION [7]. By the proof of Theorem 3, WEIGHTED-ELICITATION-NOT-DONE is equivalent to DESTRUCTIVE-COALITIONAL-WEIGHTED-MANIPULATION. ■

All of the voting protocols under study, except STV, satisfy the preconditions of the theorem. Thus:

Corollary 1 *WEIGHTED-ELICITATION-NOT-DONE is in $\mathcal{P}$ for the Borda, Copeland, Maximin, Plurality, and Approval[4] protocols.*

5 Hardness of deciding which votes to elicit

In this section we study the complexity of deciding which voters' preferences should be elicited so as to be able to determine the winning candidate while minimizing elicitation cost. In Section 5.1 we study the case where all voters have equal weight ("say-so") and equal cost of being elicited. In Section 5.2 we study the general case where the weights and elicitation costs may vary across voters.

5.1 Equal weights and equal elicitation costs

The elicitor could use its suspicions about how the agents will vote to try to design the elicitation policy so that few queries are needed. The suspicions

[4]Because the Approval protocol does not make the voters rank the candidates, Theorem 4 needs some minor modifications to cover the Approval protocol.

could be represented by a joint prior distribution over the agents' votes. It is perhaps not too surprising that in this general setting, computational complexity issues arise with regard to optimal elicitation, because the number of probabilities in a general joint prior distribution is $(m!)^n$. Given that this is an impractically large amount of information to generate (and to input into an elicitor bot), it is reasonable to presume that the language the elicitor uses to express its suspicions is not fully expressive. With such a restricted language, one might hope that the optimal elicitation problem is tractable. However, this turns out not to be the case! We show that if this language even accomodates as little as degenerate distributions (all the probability mass on a single vote), determining an optimal coarse elicitation policy is hard. In other words, it is hard even with perfect suspicions. This even holds for unweighted voters with uniform elicitation costs (as long as the number of candidates is allowed to grow). We define the effective elicitation problem with perfect suspicions as follows:

Definition 6 (EFFECTIVE-ELICITATION) *We are given a set of votes S and a number k. We are asked whether there is a subset of S of size $\leq k$ that decides the election constituted by the votes in S.*

All the reductions in this section will be from 3-COVER.

Definition 7 (3-COVER) *We are given a set U of size $3q$ and a collection of subsets $\{S_i\}_{1\leq i\leq r}$ of U (where $r > q$), each of size 3. We are asked if there is a cover of U consisting of q of the subsets.*

Theorem 5 *For the Approval protocol,* EFFECTIVE-ELICITATION *is $\mathcal{NP}$-complete.*

Proof: It is easy to show that the problem is in $\mathcal{N}P$. To show $\mathcal{N}P$-hardness, we reduce an arbitrary 3-COVER instance to the following EFFECTIVE-ELICITATION instance. The candidate set is $U \cup \{w\}$. The votes are as follows. For every S_i there is a vote approving $S_i \cup \{w\}$. Additionally, we have $r - 2q + 2$ votes approving only $\{w\}$, for a total of $2r - 2q + 2$ votes. Finally, we set $k = r - q + 2$. We claim the problem instances are equivalent. First suppose there is a 3-cover. Then we elicit all the votes that approve only w, and the votes that correspond to sets in the cover, for a total of k votes. Then w is $r-q+1$ points ahead of all other candidates, with only $r-q$

votes remaining. Hence there is an effective elicitation. On the other hand, suppose there is no 3-cover. Then eliciting k votes will always give one of the candidates in U at least 2 votes, so that w can be at most $r-q$ points ahead of this candidate. Hence, with $r-q$ votes remaining, the election cannot possibly be decided. So there is no effective elicitation. ■

Theorem 6 *For the Borda protocol,* EFFECTIVE-ELICITATION *is* $\mathcal{NP}$*-complete.*

Proof: It is easy to show that the problem is in $\mathcal{NP}$. To show $\mathcal{NP}$-hardness, we reduce an arbitrary 3-COVER instance to the following EFFECTIVE-ELICITATION instance. The candidate set is $U \cup \{w\} \cup B$ where $B = \{b_1, b_2, \dots, b_{64r^2}\}$. The votes are as follows. For each S_i there is a vote ranking the candidates $(B/2, U - S_i, B/2, S_i, w)$, where the occurrence of a set in the ranking signifies all of its elements in whichever order, and $B/2$ signifies some subset of B containing half its elements. Finally, there are $4r-2q-2$ votes that rank the candidates $(w, b_1, \dots, b_{8r^2}, u_1, \dots, u_{3q}, b_{8r^2+1}, \dots, b_{64r^2})$, and another $4r-2q-2$ that rank them $(w, b_{64r^2}, \dots, b_{56r^2+1}, u_{3q}, \dots, u_1, b_{56r^2}, \dots, b_1)$. Let $g = 8r - 4q - 4$, so that we have a total of $g + r$ votes. Also, let $l = 64r^2 + 3q$, which is the number of points a candidate gets for being in first place. Finally, we set $k = g + q$. We claim the problem instances are equivalent. First suppose there is a 3-cover. We elicit all the votes that put w on top, and the votes that correspond to sets in the cover, for a total of k votes. Even after eliciting just the ones that put w on top, w is more than $\frac{gl}{2} \geq 2rl$ (since $g \geq 4r$) points ahead of all the elements of B, and with only r votes remaining it is impossible to catch up with w for anyone in B. For a given element u of U, the votes that put w on top result in a net difference of $g(8r^2 + \frac{3}{2}q + \frac{1}{2})$ points between w and u. Of the q remaining elicited votes, precisely $q-1$ placed u ahead of half the elements of B, so the net difference in points between w and u most favorable to u arising from these would be $-(q-1)(32r^2 + 3q)$. Finally, the vote that put u below all the elements of b might contribute another -3. Adding up all these net differences, we find that w is ahead by at least $64r^3 - 64qr^2 + 12qr - 9q^2 + 4r - 5q - 5$ points. On the other hand, the maximum number of points u could gain on w with the remaining number of votes is $(r-q)(64r^2 + 3q) = 64r^3 - 64qr^2 + 3qr - 3q^2$. It is easily seen that the second expression is always smaller, and hence w is guaranteed to win the election. So there is an effective elicitation. On the other hand, suppose there is no 3-cover. First, we observe that w will always win the election - we have already shown that the votes that put w on top guarantee it does better than any element of B. For any element u of U, even if u is always placed above all the other votes in U in the r votes corresponding to the S_i, it will still only gain $r(32r^2 + 3q)$ points on w here, which is fewer than the $g(8r^2 + \frac{3}{2}q + \frac{1}{2})$ votes it loses on w with the other votes (since $g \geq 4r$). So we can only hope to guarantee that w wins. Now, if there is an elicitation that guarantees this, there is also one that elicits all the g votes that put w on top, since replacing one of the other votes with such a vote in the elicitation never hurts w's relative performance to another candidate. But in such an elicitation, there is at least one candidate u in U that is never ranked below all the elements of B in the q votes elicited that put w at the bottom, since there is no 3-cover. Let us investigate how many points w may be ahead of u after eliciting these votes. Again, the votes that put w on top result in a net difference of $g(8r^2 + \frac{3}{2}q + \frac{1}{2})$ points. In the scenario most favorable to w, u would only gain $q(32r^2 + 4)$ points with the other q votes. Adding this up, w is ahead by at most $64r^3 - 64qr^2 - 32r^2 + 12qr - 6q^2 + 4r - 6q - 2$ after the elicitation. The maximum number of points u could gain on w with the remaining number of votes is still $64r^3 - 64qr^2 + 3qr - 3q^2$. It is easily seen that the second expression is always larger, so we cannot guarantee that w wins. So there is no effective elicitation. ■

Theorem 7 *For the Copeland protocol,* EFFECTIVE-ELICITATION *is* $\mathcal{NP}$*-complete.*

Proof: It is easy to show that the problem is in $\mathcal{NP}$. To show $\mathcal{NP}$-hardness, we reduce an arbitrary 3-COVER instance to the following EFFECTIVE-ELICITATION instance. The candidate set is $U \cup \{w\}$. The votes we are given are as follows. For every S_i there is a vote which places S_i in the top three spots (in whichever order), w in the fourth spot, and the other candidates in the spots below (in whichever order). Additionally, we have $r - 2q + 3$ votes placing w in the top spot and the other candidates below it in whichever order, for a subtotal of $W = 2r - 2q + 3$ votes. Finally, we impose an arbitrary order on the elements of U and label them $a_0, a_1, \ldots, a_{3q-1}$. For each $i \in \{0, 1, 3q - 1\}$, we have another W votes which rank the candidates $(w, a_i, a_{i+1(mod 3q)}, \ldots, a_{i+3q-1(mod 3q)})$, and another W which rank them $(a_i, a_{i-1(mod 3q)}, \ldots, a_{i-3q+1(mod 3q)})$. (We call these the cyclical votes.) The total number of votes is $(6q + 1)W$. Now we set $k = (3q + \frac{1}{2})W + \frac{3}{2}$. We claim the problem instances are equivalent. First suppose there is a 3-cover. We elicit all the votes that put w on top, and the votes that correspond to sets in the cover, for a total of k votes. Then in each pairwise election involving w, we have found $k-1 = (3q+\frac{1}{2})W + \frac{1}{2}$ votes that prefer w to the other candidate. But then we have guaranteed that w wins every pairwise election, and hence the Copeland election. So there is an effective elicitation.

On the other hand, suppose there is no 3-cover. First we show that if the votes we elicit guarantee a pairwise victory between elements of U, $a_h > a_l$, they can guarantee no other pairwise victory. In order to guarantee the given pairwise victory, we need $k-1 = (3q+\frac{1}{2})W + \frac{1}{2}$ votes to prefer a_h to a_l. There are precisely $3q$ groups of cyclical votes of size W that are like this. Since there are only W noncyclical votes, it follows that we need to elicit at least $(3q-\frac{1}{2})W + \frac{1}{2}$ votes from the $3q$ cyclical groups, hence we need to elicit more than half of the votes of each of these groups. Now we claim that for each other pair of candidates c_i, c_j, at least in one of these groups all votes prefer c_i to c_j. If $c_i = w$, we take the group that starts with $(w, a_h, \ldots)$. If $c_j = w$, we take the group that ends with $(\ldots, a_l, w)$. Otherwise, we know that either the group starting $(w, a_h, \ldots)$ or the one with $(a_h, \ldots, w)$ has the desired property, since the order of the candidates represented by the dots in these in one of them is the opposite of that of the other. (Unless $c_j = a_h$; in this case either the group $(w, \ldots, a_l)$ or $(\ldots, a_l, w)$ has the desired property, because by assumption $a_l \neq c_i$ in this case.) Hence we have elicited more than $\frac{1}{2}W$ votes that prefer c_i to c_j, which means that we cannot guarantee a pairwise victory of c_j over c_i, because $k - \frac{1}{2}W$ votes are not enough for this purpose. Since these were arbitrary, we can guarantee no other pairwise victories.

Now we prove there can be no effective elicitation. First we note that we cannot guarantee that w will win every pairwise election, since we must elicit at least q votes from the votes representing the S_i; and as there is no 3-cover, this means that there is some candidate c that will be preferred to w twice in this elicitation, and $k-2$ votes is too few to guarantee the outcome of a pairwise election. Hence, if we wish to guarantee that w will win, we have to make sure that c loses at least two other elections; since otherwise, c may win all the pairwise elections that it still has a chance of winning (e.g., if all the non-elicited votes put c on top), and at least tie with w. But we have just shown that it is impossible to guarantee two pairwise victories not involving w. So we cannot guarantee that w will win. Could we perhaps guarantee that another candidate wins the election? No, since we know we could guarantee at most one pairwise victory for such a candidate. In the worst case (all the other, non-elicited votes put this candidate at the bottom), this is the only victory it will get, which cannot be enough to guarantee victory. So there is no effective elicitation. ■

Theorem 8 *For the* MAXIMIN *protocol,* EFFECTIVE-ELICITATION *is* $\mathcal{NP}$*-complete.*

Proof: It is easy to show that the problem is in $\mathcal{NP}$. To show $\mathcal{NP}$-hardness, we reduce an arbitrary 3-COVER instance to exactly the same EFFECTIVE-ELICITATION instance as we just did for Copeland. We show the two problem instances are equivalent.

First suppose there is a 3-cover. As before we can elicit votes that guarantee that w wins every pairwise election. Thus, w will get more than half of the votes in its worst pairwise election, and all the others will get fewer than half (since at least w defeated them in their pairwise election), and w is guaranteed to win. So there is an effective elicitation.

On the other hand, suppose there is no 3-cover. Then, as before, after eliciting k votes there is still some candidate c that w is not guaranteed to defeat. Hence, if we wish to guarantee that w wins, we should guarantee that c loses to someone else, or else c could at least tie with everyone and thus do at least as well in its worst pairwise election as w (since w at best ties with c). But, as we have shown in the previous proof, guaranteeing such a loss entails that we find at most $3qW + \frac{1}{2}$ votes for w in any pairwise election. Then, if some other candidate were to be placed on top, and w at the bottom, in all the other, non-elicited votes, this would give the former $(3q+\frac{1}{2})W - \frac{3}{2}$ votes in any pairwise election, more than enough to defeat w's worst score. So we cannot guarantee w wins the election. Could we perhaps guarantee that another candidate c wins the election? No - we would need to ensure that each other candidate loses at least one pairwise election. All of these guaranteed losses must somehow involve w since otherwise we can only guarantee one such loss; hence, w would have to be guaranteed to beat each other candidate except c. But since the votes that put w at the bottom are useless for this purpose, we cannot elicit more than one of these; and hence $k - r - 1$ of the votes we elicit must be from the ones that put w on top. But then it is impossible to guarantee that w has a pairwise loss. So there is no effective elicitation. ■

So far in this section we have shown that determining an effective elicitation policy is hard for all of the protocols under study, except for STV and Plurality. From the previous section, we already know that for the STV protocol even knowing when to terminate is hard. The remaining protocol is Plurality, where it is easy to elicit effectively given perfect suspicions (start eliciting the winner's votes first; if all of them have been elicited and termination is still not possible, elicit votes in a round-robin manner, one for each non-winning candidate (as long as it has votes left), until the elicitation can terminate).

In this section we showed that effective elicitation is $\mathcal{N}P$-complete for various voting protocols even when the elicitor has perfect suspicions about how the voters will vote. Naturally, the elicitation problem does not become easier if the elicitor has only probabilistic information about how the voters will vote. In the appendix we show that in that setting, effective elicitation can even be $\mathcal{P}SPACE$-hard.

5.2 Unequal weights or elicitation costs

We now analyze the general case where the voters' weights or the costs of eliciting their preferences may differ across voters. All the hardness results from Section 5.1 still hold in all of these settings because hardness results carry over from special settings to more general settings. We now show, however, that weights may introduce complexity even in the case where the number of candidates is constant (even just 2).

Definition 8 (WEIGHTED-EFFECTIVE-ELICITATION) *We are given a set of votes S with elicitation costs c_i and weights w_i, and a number B. We are asked whether there is a subset of S with combined elicitation cost $\leq B$ that decides the election constituted by the votes in S.*

In order to demonstrate $\mathcal{NP}$-hardness, we reduce from the PARTITION problem.

Definition 9 (PARTITION) *We are given a set of integers $\{k_i\}_{1\leq i\leq t}$ (possibly with multiplicities) summing to $2K$, and are asked whether a subset of these integers sum to K.*

We are now ready to state our result. If there are only two candidates, all of the voting protocols under study coincide to a "standard" protocol: the candidate with more votes wins.[5]

Theorem 9 *For the* 2*-candidate "standard" voting protocol,* WEIGHTED-EFFECTIVE-ELICITATION *is $\mathcal{NP}$-complete, even when for all i, $c_i = w_i$ or $c_i = 0$.*

Proof: It is easy to show that the problem is in $\mathcal{N}P$. To show $\mathcal{N}P$-hardness, we reduce an arbitrary PARTITION instance to the following WEIGHTED-EFFECTIVE-ELICITATION instance. The candidates are a and b. For each k_i let there be a voter with $c_i = w_i = k_i$, voting for candidate a. Additionally, let there be a single voter with $c_0 = 0$, $w_0 = 1$, also voting for candidate a. Let $B = K$. We claim the instances are equivalent.

First suppose there is a solution to the PARTITION instance, that is, a subset of the integers summing to K. Then eliciting the corresponding voters, and the single costless voter, has a total cost of precisely K, and demonstrates that $K + 1$ of the total vote weight of $2K + 1$ votes for candidate a. So there is a solution to the WEIGHTED-EFFECTIVE-ELICITATION instance.

On the other hand, suppse there is a solution to the WEIGHTED-EFFECTIVE-ELICITATION instance. Consider the set of k_i corresponding to elicited voters. By the cost constraint, these k_i can sum to at most K. Since the election is decided by the elicted votes, and the weight of the elicited votes not corresponding to k_i can be at most 1, it follows that the subset of k_i corresponding to elicited voters must sum to at least K. But then this subset must sum to K exactly. So there is a solution to the PARTITION instance. ■

Because all the voting protocols under discussion reduce to the "standard" voting protocol when there are two candidates, we can conclude the following:

Corollary 2 *For all of the protocols studied in this paper,* WEIGHTED-EFFECTIVE-ELICITATION *is $\mathcal{NP}$-complete, even when for all i, $c_i = w_i$ or $c_i = 0$, and the number of candidates is* 2.

[5]The exception is the Approval protocol where a voter can approve or disapprove both candidates. However, Theorem 9 applies to that protocol as well because it is possible that each voter only approves one candidate. (Also, approving or disapproving both candidates is nonsensical because it will not affect the outcome of the election.)

However, if either elicitation costs or weights are equal across voters, and checking for termination is easy, there exist algorithms for WEIGHTED-EFFECTIVE-ELICITATION that are exponential only in the number of candidates, implying that the problem is not hard for any constant number of candidates. (And, by the results of Section 5.1, we already know that the WEIGHTED-EFFECTIVE-ELICITATION problem in these cases is hard when the number of candidates is allowed to grow.)

Theorem 10 *If $w_i = 1$ for all i, then for all voting protocols where the voters do not have a richer voting language than ranking the candidates*[6]*, WEIGHTED-EFFECTIVE-ELICITATION can be solved by checking for $O((n+1)^{m!})$ (or $O((n+1)^{2^m})$ in the Approval protocol) possible elicitations of size k whether the process could be terminated after this elicitation.*

Proof: The key observation is that when the weights are all equal, there is no reason to ever elicit a voter with greater elicitation cost instead of another voter with smaller elicitation cost that casts the same vote. Hence, we divide the voters into $m!$ pools (or 2^m in the Approval protocol) on the basis of the vote they will cast; then we order the voters in each pool by decreasing cost. By the above observation, we know that if there exists an effective elicitation, there also exists one where in each pool p, we only elicit the cheapest k_p voters, for some vector of k_p values. Since there are at most $n+1$ possibilities for each k_p, it follows that the total number of elicitations we need to check is $O((n+1)^{m!})$ (or $O((n+1)^{2^m})$ in the Approval protocol). ■

In the case where the elicitaton costs are equal across voters, the easiness result holds under the *block-vote termination property*:

Definition 10 *A voting protocol satisfies the* block-vote termination property *if, given a set S of known votes, a set T of voters whose votes are not yet known, and a candidate h, there exists a way to cast the votes in T to make h not win if and only if the voters in T can make h not win by all casting the same vote (hence a "block vote").*

It follows from Theorem 4 that all the voting protocols studied in this paper, except STV, satisfy the block-vote termination property.

We will make use of the following lemma in our next theorem.

[6]This result actually applies to all voting protocols, but the $O()$ complexity may differ.

Lemma 1 *Consider the following two scenarios. In the first, we are given a set S of known votes, a set T of voters whose votes are not yet known, and a candidate h. The second is the same except T is replaced by a set U of $\sum_{t \in T} w_t$ votes with weights 1 (that is, we can cast every unit of vote weight independently). Then, if the protocol satisfies the block-vote termination property, there exists a way to cast the votes in T to make h not win if and only if there exists a way to cast the votes in U to make h not win.*

Proof: By the block-vote termination property, it is possible to make h not win in both cases if and only there is a block vote of weight $\sum_{t \in T} w_t = |U|$ that makes h not win. ■

Now we are ready to present our result for equal elicitation costs and unequal weights.

Theorem 11 *Consider any voting protocol where the voters do not have a richer voting language than ranking the candidates. If $c_i = 1$ for all i, and the protocol satisfies the block-vote termination property, then WEIGHTED-EFFECTIVE-ELICITATION can be solved by checking for $O((n+1)^{m!})$ (or $O((n+1)^{2^m})$ in the Approval protocol) possible elicitations of size k whether the process could be terminated after this elicitation.*

Proof: The key observation here is that when the elicitation costs are all equal, and the block-vote termination property holds, there is no reason to ever elicit a voter with smaller weight instead of another voter with greater weight that casts the same vote. (If, in the former case where we actually elicit a vote with a smaller weight, we can terminate elicitation, it follows from Lemma 1 that there is no way to cast the remaining votes to change the winner even if we could cast each unit of vote weight independently. But then, if we elicit the vote with greater weight instead, this essentially corresponds to specifying some of the remaing vote weight, which certainly cannot help in changing the winner of the election.)

Hence, we divide the voters into $m!$ pools (or 2^m in the Approval protocol) on the basis of the vote they will cast; then we order the voters in each pool by increasing weight. By the above observation, we know that if there exists an effective elicitation, there also exists one where in each pool p, we only elicit the k_p voters with the greatest weight, for some vector of k_p values. Since there are at most $n+1$ possibilities for each k_p, it follows that the total number of elicitations we need to check is $O((n+1)^{m!})$ (or $O((n+1)^{2^m})$ in the Approval protocol). ■

(More precisely, because we know we will elicit (no more than) k voters in this case, the number of possible elicitations that we need to check is at most $\binom{m!+k-1}{m!-1}$—k indistinguishable balls into $m!$ distinguishable bins.)

So, effective elicitation with constant elicitation costs but varying weights is easy for the Approval, Plurality, Borda, Copeland, and Maximin protocols. For STV with varying weights, we have already shown that even termination is hard.

6 Strategy-proofness of elicitation

We now turn to strategic issues that may be introduced into a voting protocol by an elicitation process. Elicitation may reveal information about other agents' votes to an agent, which the agent may use to change its vote strategically. This is undesirable for two reasons. First, it gives agents that are elicited later an unfair advantage, causing the protocol to put undue weight on their preferences. Second, it leads to less truthful voting by the agents. This is undesirable because, while voting protocols are designed to select a socially desirable candidate if agents vote truthfully, untruthful voting can lead to a reduction in the social desirability of the outcome. We demonstrate how such strategic issues may arise, and then suggest avenues to circumvent them. However, these avenues entail restricting the space of possible elicitations, causing a reduction in the potential savings from elicitation.

To analyze strategic interactions, we need some tools from game theory. To bring the voting setting into the framework of noncooperative game theory, we assume that agent i's preferences are defined by its *type* θ_i; the agent gets utility $u_i(\theta_i, c)$ if candidate c wins. We first define a game:

Definition 11 *In a* (normal form) game*, we are given a set of agents A; a set of types Θ_i for each agent i; a commonly known prior distribution ϕ over $\Theta_1 \times \Theta_2 \times \ldots \times \Theta_{|A|}$; a set of strategies Σ_i for each agent $i \in A$; a set of outcomes O (candidates in the case of voting); an outcome function $o : \Sigma_1 \times \Sigma_2 \times \ldots \times \Sigma_{|A|} \rightarrow O$; and a utility function $u_i : \Theta_i \times O \rightarrow \Re$ for each agent $i \in A$.*

An agent knows its own type and can thus let its strategy depend on its type according to a function $f_i : \Theta_i \to \Sigma_i$. We also need a notion of how an agent would play a game strategically. This may depend on how others play.

Definition 12 *A strategy function profile* $(f_1, f_2, \ldots, f_{|A|})$ *is a* Bayes-Nash equilibrium (BNE)*, if for each agent* $i \in A$*, each* $\theta_i \in \Theta_i$*, and each strategy* $\sigma_i \in \Sigma_i$*,*

$$E_\phi(u_i(\theta_i, o(f_1(\theta_1), f_2(\theta_2), \ldots, f_i(\theta_i), \ldots, f_{|A|}(\theta_{|A|})))|\theta_i)$$
$$\geq$$
$$E_\phi(u_i(\theta_i, o(f_1(\theta_1), f_2(\theta_2), \ldots, \sigma_i, \ldots, f_{|A|}(\theta_{|A|})))|\theta_i)$$

(that is, each f_i *chooses, for each* θ_i*, a strategy that maximizes i's expected utility given the other players'* f_j*s).*

We are now ready to state our results.

6.1 Coarse elicitation

First we show that coarse elicitation may lead to strategic manipulations when it reveals even slightly more than just the fact that the agent is being elicited.

Theorem 12 *In a coarse elicitation protocol, the following properties can hold simultaneously:*

- *the protocol reveals no information to any agent except that the agent's type is elicited, and how many other agents have had their types elicited before,*
- *the elicitation policy is optimized to finish as quickly as possible on average given the distribution over the agents' types (presuming the agents vote truthfully), and*
- *truthful voting is a BNE with full elicitation,*
- *truthful voting is not a BNE here. In particular, an agent may have an incentive to vote differently depending on how the other agents vote.*

Proof: Consider an Approval[7] election with 3 voters, i, j and k, and 3 candidates, a, b, and c. Ties are broken randomly. Define truth-telling to mean approving all candidates that give you utility $\geq \frac{1}{2}$. Ties are broken randomly. Agents' types are independent and the distributions are as follows. With probability $\frac{1}{2}$, i has utility 1 for c, and utility 0 for a and b; with probability $\frac{1}{2}$, it has utility 1 for a, and utility 0 for b and c. With probability $\frac{1}{2}$, j has utility 1 for c, and utility 0 for a and b; with probability $\frac{1}{2}$, it has utility 1 for b and c, and utility 0 for a. It is easy to see that truth-telling is always an optimal strategy for i and j. With probability 1, k has utility 1 for a, $\frac{1}{4}$ for b, and 0 for c. For k not to approve c, and to approve a, is always optimal. In the full elicitation case, should k approve b? If j has its first type, it makes no difference. What if j has its second type? If i has its first type, approving b leads to a tie between b and c, and (expected) utility $\frac{1}{8}$; not approving b leads to a victory for c and utility 0. If i has its second type, approving b leads to a tie between a and b and utility $\frac{5}{8}$; not approving b leads to a victory for a and utility 1. Hence, in the full elicitation case, given that we are in a case where it matters whether k approves b, approving b gives utility $\frac{3}{8}$, and not approving b gives utility $\frac{1}{2}$; so not approving b is optimal. Thus, truth-telling is a BNE here.

For the coarse elicitation case, we first design a policy that is optimal with respect to the agents' type distributions. Query $Q(l)$ asks voter l which candidates it approves. Then an optimal elicitation protocol is

1: first ask $Q(i)$;

2a: if the answer was $\{c\}$, ask $Q(j)$;

2b: otherwise, ask $Q(k)$;

3: if do not know the winner yet, query the last voter.

To show optimality with respect to the type distribution, assume the agents reply truthfully. If i has its first type, and j its first, we finish after *2a*, in 2 steps. If i has its second type, we finish after *2b*, in 2 steps. But these are the only cases in which we can hope to finish in only two steps, so the protocol is optimal.

Now, if k is queried second, this implies to it that i is of its second type, and it is motivated to answer truthfully. But if i is queried third, this implies to it that i is of its first type, and j of its second type; and k is motivated to

[7] We use the Approval protocol to demonstrate the negative results (Theorems 12 and 14) because 1) this demonstrates that these strategic issues can occur even in a very simple protocol, and 2) the protocol has a natural query type also for fine elicitation. However, similar strategic issues arise in any reasonable protocol.

lie and approve b. So truth-telling is not a BNE here. ■

However, if the elicitation reveals no information to the agent being elicited (beyond the fact that the agent is being elicited), then elicitation does not introduce strategic issues:

Theorem 13 *Consider a coarse elicitation protocol which manages to reveal nothing more to the agent than whether or not his type is elicited. Then, the set of BNEs is the same as in the corresponding full elicitation voting game.*[8]

Proof: We claim that the normal form of the game is identical to that in the full elicitation setting; this implies the theorem. Obviously, the Θ_i, the u_i, and ϕ remain the same. Now consider the Σ_i. Because no information is revealed upon elicitation, the voter cannot condition its response on anything but its type, as in the full elicitation case. That is, each agent need only decide on the one vote that it will always cast if it is elicited. Hence, the strategy set of an agent is simply the space of votes, as it is in the full elicitation case. Finally, by our requirement that this elicitation produces the same outcome as full elicitation, o is the same. ■

It is an interesting open problem how to design an elicitation protocol that reveals no information about how many agents have had their types elicited so far. This seems difficult because any protocol will at least betray the real time at which an agent is queried.

6.2 Fine elicitation

We now show that fine elicitation can lead to additional strategic issues even if no unnecessary information is revealed to the agents.

Theorem 14 *In a fine elicitation protocol, the following properties can hold simultaneously:*

- *the protocol reveals no information to any agent except the queries to the agent and the order of those queries,*

[8]For the game-theoretically inclined, we observe that some of the BNEs in the coarse elicitation case are not subgame perfect. These equilibria are unstable in the full elicitation case as well.

- *the elicitation policy is optimized to finish as quickly as possible on average given the distribution over the agents' types (presuming the agents vote truthfully), and*

- *truthful voting is a BNE with full elicitation,*

- *truthful voting is not a BNE here. In particular, an agent may have an incentive to vote differently depending on how the other agents vote.*

Proof: Consider an Approval election with 2 voters, i and j, and 3 candidates, a, b, and c. Define truth-telling to mean approving all candidates that give you utility $\geq \frac{1}{2}$. Ties are broken randomly. Agents' types are independent and the distributions are as follows. With probability $\frac{1}{2}$, i has utility 1 for b and c, and utility 0 for a; with probability $\frac{1}{2}$, it has utility 1 for a and b, and utility 0 for c. It is easy to see that truth-telling is always an optimal strategy for i. With probability 1, j has utility 1 for a, $\frac{3}{4}$ for b, and 0 for c. For j not to approve c, and to approve a, is always optimal. In the full elicitation case, should j approve b? If i has its first type, approving b leads to victory for b and a utility of $\frac{3}{4}$; not approving b leads to a 3-way tie and utility of $\frac{7}{12}$. If i has its second type, approving b leads to a 2-way tie between a and b and utility $\frac{7}{8}$; not approving b leads to a victory for a and utility 1. Hence, in the full elicitation case, approving b gives utility $\frac{13}{16}$, and not approving b gives utility $\frac{19}{24}$; so approving b is optimal. Thus, truth-telling is a BNE here.

For the fine elicitation case, we first design a policy that is optimal with respect to the agents' type distributions. The natural restriction here is to allow only the following type of query: query $Q(k, d)$ asks voter k if it approves candidate d. Then an optimal elicitation protocol is

1: first ask $Q(i, a)$;

2a: if the answer was 'no', ask $Q(i, b)$; $Q(j, b)$; $Q(j, c)$;

2b: otherwise, ask $Q(j, a)$; $Q(i, b)$; $Q(j, b)$; $Q(i, c)$.

3: if we do not know the winner yet, ask the remaining queries.

To show optimality with respect to the type distribution, assume the agents reply truthfully. If i has its first type, we finish after *2a*, in 4 steps; if i has its second type, we finish after *2b*, in 5 steps. This is optimal.

Now, if the first query to j is $Q(j, b)$, this implies to it that i is of its first type and it is motivated to answer truthfully. But if the first query to j is $Q(j, a)$, this implies to it that i is of its second type; and i is motivated to lie and not approve b. So truth-telling is not a BNE here. ■

Finally, we show that with a certain restriction on elicitation policies, we can guarantee that fine elicitation does not introduce any strategic effects.

Definition 13 *A fine elicitation policy is* nondivulging *if the next query to an agent (if it comes) depends only on that agent's own responses to previous queries. (Whether or not the next query is asked can depend on the agent's* and the other agents' *responses to queries so far.)*

Theorem 15 *Consider a fine elicitation protocol which manages to reveal nothing more to the agent than the queries to the agent and the order of those queries. If the elicitation policy is nondivulging, then the set of BNEs is the same as in the full elicitation voting game.*

Proof: We claim that the normal form of the game is identical to that in the full elicitation setting; this implies the theorem. Obviously, the Θ_i, the u_i, and ϕ remain the same. Now consider the Σ_i. Because the agent knows the first query to it (if it comes), it can determine its response up front. The next query (if it comes) can only depend on this response, so the agent knows it, and can prepare a response to it up front as well; and so on. So, in this setting, we can define the agent's strategy to be this entire sequence of responses. But this sequence correponds to exactly one vote in the full elicitation case.[9] Hence, the strategy set of an agent is simply the space of votes, as it is in the full elicitation case. Finally, by our requirement that this elicitation produces the same outcome as full elicitation, o must be the same. ■

While a restriction to nondivulging elicitation policies avoids introducing additional strategic effects, it can reduce the efficiency of elicitation.

7 Conclusion and future research

Preference elicitation is a central problem in AI, and has received significant attention in single-agent settings. Preference elicitation is also a key problem in multiagent systems, but has received little attention here. In this setting,

[9] By our definition of a fine elicitation policy, no queries are asked that would enable an agent to express inconsistent (e.g., cyclical) preferences.

the agents may have different preferences that often must be aggregated using voting. This leads to interesting issues because what, if any, information should be elicited from an agent depends on what other agents have revealed about their preferences so far. In this paper we studied effective elicitation for the most common voting protocols: Plurality, Approval, Borda, Copeland, Maximin, and Single-Transferable Vote (STV).

We first studied the complexity of determining whether enough information has been elicited. We showed that this is $\mathcal{NP}$-complete in the STV protocol (even with unweighted voters or, if we allow for weighted voters, even with just 4 candidates). We presented a polynomial-time termination algorithm for elicitation in the STV protocol when the number of candidates is constant and the voters are unweighted. For all the other protocols under study, we presented a polynomial-time termination algorithm that applies even with weighted voters and an unbounded number of candidates.

For protocols other than STV, we studied the complexity of deciding which votes to elicit. When voters are equally weighted and have the same elicitation costs, then for an unbounded number of candidates, for each of these protocols, determining how to elicit effectively is $\mathcal{NP}$-complete even with perfect suspicions about how the agents will vote. The exception is the Plurality protocol where such effective elicitation is easy. When voters are weighted and elicitation costs may vary across voters, effective elicitation is $\mathcal{NP}$-complete for all voting protocols, even with just two candidates and perfect suspicions about how the voters will vote. For unweighted voters with varying elicitation costs and perfect suspicions, we developed an optimal elicitation algorithm that is exponential only in the number of candidates (implying that the problem is easy for any constant number of candidates). For weighted voters with uniform elicitation costs (with perfect suspicions), for voting protocols that have the *block-vote termination* property (including all of the protocols studied in this paper except STV), we developed an optimal elicitation algorithm that is exponential only in the number of candidates (implying that the problem is easy for any constant number of candidates). Finally, we showed that if the elicitor is uncertain about how the voters will vote, effective elicitation can become $\mathcal{PSPACE}$-hard (even with unweighted voters and uniform elicitation costs, but an unbounded number of candidates).

Our results on strategy-proofness showed that elicitation can introduce opportunities for strategic manipulation of the election by the voters—beyond the manipulation opportunities present without elicitation. This is the case even with coarse elicitation (where a voter's vote is elicited entirely or not at all), if a voter can infer how many other voters were elicited before it. We also showed that if voters cannot infer this, coarse elicitation introduces no manipulation possibilities. On the other hand, we showed that fine elicitation (where a vote can be elicited partially and incrementally) can introduce manipulation opportunities even when no voter can infer how many voters were elicited before it. Finally, we showed that if the fine elicitation queries are always asked in a fixed order (for a given voter), elicitation introduces no manipulation opportunities.

Future research includes studying elicitation policies that choose the right outcome with high *probability* rather than with certainty. It also includes designing new voting protocols that combine the computational ease of elicitation in the Plurality protocol with the expressiveness of the other protocols. Finally, it would be interesting to study specific fine elicitation schemes in more detail, and design elicitation protocols that reveal no information about how many other agents have been elicited so far.

References

[1] John J. Bartholdi, III and James B. Orlin. Single transferable vote resists strategic voting. *Social Choice and Welfare*, 8(4):341–354, 1991.

[2] Craig Boutilier, Ronen Brafman, Christopher Geib, and David Poole. A constraint-based approach to preference elicitation and decision making. In *AAAI Spring Symposium on Qualitative Decision Theory*, 1997.

[3] U Chajewwska, Daphne Koller, and R Parr. Making rational decisions using adaptive utility elicitation. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 363–369, Austin, TX, 2000.

[4] Wolfram Conen and Tuomas Sandholm. Preference elicitation in combinatorial auctions: Extended abstract. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, pages 256–259, Tampa, FL, October 2001. A more detailed description of the algorithmic aspects appeared in the IJCAI-2001 Workshop on Economic Agents, Models, and Mechanisms, pp. 71–80.

[5] Wolfram Conen and Tuomas Sandholm. Differential-revelation VCG mechanisms for combinatorial auctions. In *AAMAS-02 workshop on Agent-Mediated Electronic Commerce (AMEC)*, Bologna, Italy, 2002.

[6] Wolfram Conen and Tuomas Sandholm. Partial-revelation VCG mechanism for combinatorial auctions. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, Edmonton, Canada, 2002.

[7] Vincent Conitzer and Tuomas Sandholm. Complexity of manipulating elections with few candidates. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, Edmonton, Canada, 2002.

[8] Joseph D'Ambrosio and William Birmingham. Preference-directed design. *Artificial Intelligence in Engineering Design, Analysis and Manufacturing*, 1995.

[9] Carmel Domschlak, Ronen Brafman, and Solomon Shimony. Preference-based configuration of web page content. In *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1451–1456, Seattle, WA, 2001.

[10] Marek Druzdel and Linda van der Gaag. Elicitation of probabilities for belief networks: Combining qualitative and quantitative information. In *Proceedings of the Uncertainty in Artificial Intelligence Conference (UAI)*, 1995.

[11] Eithan Ephrati. A non-manipulable meeting scheduling system. In *Proc. 13th International Distributed Artificial Intelligence Workshop*, Lake Quinalt, Washington, July 1994. AAAI Press Technical Report WS-94-02.

[12] Eithan Ephrati and Jeffrey S Rosenschein. The Clarke tax as a consensus mechanism among automated agents. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 173–178, Anaheim, CA, 1991.

[13] Eithan Ephrati and Jeffrey S Rosenschein. Multi-agent planning as a dynamic search for social consensus. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 423–429, Chambery, France, 1993.

[14] Eithan Ephrati and Jeffrey S Rosenschein. Deriving consensus in multi-agent systems. *Artificial Intelligence*, 87(1–2):21–74, 1996.

[15] Eithan Ephrati and Jeffrey S Rosenschein. A heuristic technique for multi-agent planning. *Annals of Mathematics and Artificial Intelligence*, 20:13–67, 1997.

[16] Benoit Hudson and Tuomas Sandholm. Effectiveness of preference elicitation in combinatorial auctions. In *AAMAS-02 workshop on Agent-Mediated Electronic Commerce (AMEC)*, Bologna, Italy, 2002. Extended version: Carnegie Mellon University, Computer Science Department, CMU-CS-02-124, March. Also: Stanford Institute for Theoretical Economics workshop (SITE-02).

[17] Vijay Iyengar, Jon Lee, and Murray Campbell. Q-Eval: Evaluating multiple attribute items using queries. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, pages 144–153, Tampa, FL, 2001.

[18] Kate Larson and Tuomas Sandholm. Bargaining with limited computation: Deliberation equilibrium. *Artificial Intelligence*, 132(2):183–217, 2001. Short early version appeared in the Proceedings of the National Conference on Artificial Intelligence (AAAI), pp. 48–55, Austin, TX, 2000.

[19] C Papadimitriou. Games against nature. *Journal of Computer and System Sciences*, 31:288–301, 1985.

[20] David C Parkes. Optimal auction design for agents with hard valuation problems. In *Agent-Mediated Electronic Commerce Workshop at the International Joint Conference on Artificial Intelligence*, Stockholm, Sweden, 1999.

[21] David C Parkes and Lyle Ungar. Iterative combinatorial auctions: Theory and practice. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 74–81, Austin, TX, August 2000.

[22] David C Parkes and Lyle Ungar. Preventing strategic manipulation in iterative auctions: Proxy-agents and price-adjustment. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 82–89, Austin, TX, August 2000.

[23] T L Saaty. *The Analytical Hierarchy Process*. McGraw Hill, 1980.

[24] Ahti Salo and Raimo Hämäläinen. Preference ratios in multiattribute evaluation (PRIME): Elicitation and decision procedures under incomplete information, 2001.

[25] Tuomas Sandholm. An implementation of the contract net protocol based on marginal cost calculations. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 256–262, Washington, D.C., July 1993.

[26] Tuomas Sandholm and Fredrik Ygge. Constructing speculative demand functions in equilibrium markets. Technical Report WUCS-99-26, Washington University, Department of Computer Science, October 1999. Short early version appeared at the International Joint Conference on Artificial Intelligence (IJCAI), pp. 632–638, Nagoya, Japan, 1997.

[27] Trey Smith, Tuomas Sandholm, and Reid Simmons. Constructing and clearing combinatorial exchanges using preference elicitation. In *AAAI-02 workshop on Preferences in AI and CP: Symbolic Approaches*, 2002.

[28] Ha Vu and Peter Haddawy. Problem-focused incremental elicitation of multi-attribute utility models. In *Proceedings of the Uncertainty in Artificial Intelligence Conference (UAI)*, pages 215–222, San Francisco, CA, 1997.

[29] Ha Vu and Peter Haddawy. Towards case-based preference elicitation: Similarity measures on preference structures. In *Proceedings of the Uncertainty in Artificial Intelligence Conference (UAI)*, pages 193–201, 1998.

[30] Michael Wellman. A computational market model for distributed configuration design. In *Proc. 12th National Conference on Artificial Intelligence (AAAI-94)*, pages 401–407, Seattle, WA, July 1994.

[31] Peter R Wurman and Michael P Wellman. AkBA: A progressive, anonymous-price combinatorial auction. In *Proceedings of the ACM Conference on Electronic Commerce (ACM-EC)*, pages 21–29, Minneapolis, MN, October 2000.

A PSPACE-hardness of elicitation under imperfect suspicions

In this section, we demonstrate that if we do not know how the voters will vote in advance, even stronger measures of computational hardness may apply to the elicitation problem. Specifically, we show that in this case, elicitation in the Approval protocol is $\mathcal{PSPACE}$-hard, even with unweighted voters with uniform elicitation cost. (As in the case where we knew the votes in advance, this proof may be used as a template for proving $\mathcal{PSPACE}$-hardness of elicitation in other voting protocols, though we will not present such modifications of the proof here.) At the end of this appendix, we discuss why we believe that the hardness results for the case where the votes are known in advance are more interesting.

Since we no longer know the votes in advance, it is no longer appropriate to phrase the elicitation problem in terms of selecting a subset of the votes. Rather, we investigate the existence of desirable *elicitation trees*, that is, contingency plans (online control policies) for eliciting. Additionally, it is no longer appropriate to require that we know the outcome with certainty after eliciting k votes; rather, we require that we know the outcome with high probability. The precise definition of this extended elicitation problem follows.

Definition 14 (EFFECTIVE-ELICITATION-WITH-UNCERTAINTY (EEWU)) *We are given a set of voters S; for each voter s_i in S, a probability distribution Γ_i over all possible votes; an integer k; and a number r ($0 \leq r \leq 1$). We are asked whether there exists a contingency plan (elicitation tree) for eliciting the votes such that with probability at least r, we will know the winner of the election after eliciting (at most) k votes.*

A general probability distribution Γ_i may require exponential space to represent. One way to prevent this is by requiring that the support of each distribution Γ_i be polynomial in size. We show hardness even when the support has size at most 3.[10]

[10] As before, we do not allow the elicitor to terminate elicitation on the basis of knowledge of the support of a voter's distribution (as far as terminating elicitation is concerned, each unelicited voter can still cast any of all the possible votes). In other words, as in the case of perfect suspicions, termination is not allowed based on suspicions implying an event has probability 0—rather the elicitor has to be able to produce a certificate to an outsider,

In order to demonstrate $\mathcal{PSPACE}$-hardness, we reduce from the $\mathcal{PSPACE}$-complete *stochastic satisfiability* problem [19].

Definition 15 (STOCHASTIC-SAT (SSAT)) *We are given a Boolean formula ϕ in conjunctive normal form (represented by a set of clauses Cl, over variables $X \cup Y = \{x_1, x_2, \ldots, x_q, y_1, y_2, \ldots, y_q\}$). We play the following game with nature: we pick a value for x_1, subsequently nature randomly picks a value y_1, whereupon we pick a value for x_2, after which nature randomly picks a value for y_2, etc., until all variables have a value. We are asked whether there is a policy (contingency plan) for playing this game such that the probability of the formula being eventually satisfied is at least $\frac{1}{2}$.*

We are now ready to state our result.

Theorem 16 *In the Approval protocol, EEWU is $\mathcal{PSPACE}$-hard (even if no distribution ranges over more than 3 votes, and the probability of a particular vote must be either 0, $\frac{1}{4}$, $\frac{1}{2}$, or 1).*

Proof: We reduce an arbitrary SSAT instance to the following EEWU instance. We are given a Boolean formula in conjunctive normal form with a set of clauses Cl, over variables $X \cup Y = \{x_1, x_2, \ldots, x_q, y_1, y_2, \ldots, y_q\}$. Let $W_i = \{w_{i,i+1}, \ldots, w_{i,q}\}$. Then, let $W = \bigcup_{1 \leq i \leq q} W_i$. ($W$ consists of the elements of an upper triangular matrix, without the diagonal.) Now let the candidate set be $C = \{p\} \cup Cl \cup X \cup W$. Let $sat(l)$ be the set of clauses in Cl that is satisfied by the literal l, e.g., $sat(-x_1) = \{cl \in Cl : -x_1 \in cl\}$. Let there be the following voters:

- $|W| + q = \frac{q(q-1)}{2} + q$ voters that approve only p with probability 1;
- For each $w_{i,j} \in W$, a voter $v_{w_{i,j}}$ that approves $C - W_j - \{w_{i,j}, x_j\}$ with probability 1, for a total of $|W| = \frac{q(q-1)}{2}$ voters;
- For each x_j, a voter v_{x_j} that approves $C - \{x_j\} - W_j - sat(x_j) - sat(y_j)$ with probability $\frac{1}{4}$; $C - \{x_j\} - W_j - sat(x_j) - sat(-y_j)$ with probability $\frac{1}{4}$; and $C - Cl - x_j$ with probability $\frac{1}{2}$, for a total of q voters;
- For each x_j, a voter v_{-x_j} that approves $C - \{x_j\} - W_j - sat(-x_j) - sat(y_j)$ with probability $\frac{1}{4}$; $C - \{x_j\} - W_j - sat(-x_j) - sat(-y_j)$ with probability $\frac{1}{4}$; and $C - Cl - x_j$ with probability $\frac{1}{2}$, for a total of q voters.

proving from the elicited votes that the elicitation is done (when the outsider does not know anything about the voters).

Hence, the total number of voters is $2|W|+3q = q(q+2)$. Let $k = |W|+2q = \frac{q(q+3)}{2}$. Finally, let $r = 1 - (\frac{1}{2})^{q+1}$.[11] We claim that the EEWU instance has a solution if and only if the SSAT instance has a solution.

First suppose there is a solution to the SSAT instance, that is, a contingency plan for setting the x_i such that ϕ will be satisfied with probability at least $\frac{1}{2}$. Then consider the following contingency plan for the EEWU instance:

- First we elicit all the voters that approve only p;
- Then, as long as none of the voters corresponding to the x_j or $-x_j$ have turned out to vote according to their third possible vote (the one with probability $\frac{1}{2}$), we follow the contingency plan that solves the SSAT instance. That is, if the SSAT contingency plan sets x_1 to *true*, we elicit v_{x_1} first, otherwise v_{-x_1}; then, depending on whether the vote turned out to not approve $sat(y_1)$ or $sat(-y_1)$, we say that y_1 has been "set" to *true* or *false*, respectively; then we look at the SSAT contingency plan to see how x_2 should be set in this case, and elicit v_{x_2} or v_{-x_2} accordingly; and so on.
- On the other hand, if voter v_{x_i} or voter v_{-x_i} turns out to vote according to the third possible vote, then from here on we elicit the voters corresponding to elements of W_i (that is, voters $v_{w_{i,i+1}}, \ldots, v_{w_{i,q}}$) in order.

Note that this indeed elicits precisely $|W| + 2q = k$ voters. We now proceed to analyze the probability that we can terminate elicitation at the end of this. After eliciting all the voters that approve only p, p will be ahead of the other candidates by $|W| + q$ votes. Because everyone always approves p, it follows that we can terminate elicitation after k elicitations if and only if each candidate besides p is not approved by at least one of the remaining q elicited voters. (Because this, and only this, will put p ahead of all other candidates by at least $|W|+q+1$ votes, and after the k elicitations there are still $|W|+q$ voters remaining.) Now, it is easy to verify that the $|W| + q + i$th elicited vote will not approve x_i; also, the $|W|+q+i$th elicited vote will not approve any of W_i, unless this voter casts its third possible vote, in which case each $w_{i,j} \in W_i$ will remain unapproved by the $|W| + q + j$th elicited voter, $v_{w_{i,j}}$.

[11] Note that even though we are using q in the exponent, our reduction is nevertheless polynomial in size, since the length of the binary representation of r is linear in q.

It follows that the only other candidates that may be approved by all the remaining q elicited votes are the ones in Cl. There is no possibility that one of these is approved by all the remaining votes, unless none of the elicited voters vote according to their third type; this happens with probability $(\frac{1}{2})^q$. Given that this happens, it is easy to show that the $|W|+q+i$th elicited voter votes according to its first type with probability $\frac{1}{2}$ and according to its second type with probability $\frac{1}{2}$; and all of these events are independent. Now we observe that when we elicit v_{x_i} (v_{-x_i}), this has the effect of eliminating all the candidates in $sat(x_i)$ ($sat(-x_i)$); and the $|W|+q+i$th elicited voter voting according to the first (second) possible vote, which corresponds to "setting" y_i to *true* (*false*), has the effect of eliminating $sat(y_i)$ ($sat(-y_i)$). Because, as long as no elicited voter votes according to its third possible vote, we follow the contingency plan satisfying the SSAT instance, it follows from the above that in this setting we eliminate all clauses with probability at least $\frac{1}{2}$. We conclude that the probability that we cannot terminate the elicitation after k elicitations is at most $(\frac{1}{2})^{q+1}$, hence the probability of success is at least $1-(\frac{1}{2})^{q+1}=r$. So there exists a solution to the EEWU instance.

Conversely, suppose there exists a solution to the EEWU instance. Consider a contingency plan that maximizes the probability that we can terminate elicitation after k elicitations. It never hurts to elicit the votes that approve only p, so we may assume that the contingency plan first elicits all of these. Also, since eliciting one of the $v_{w_{i,j}}$ reveals no information, by the principle of least commitment, we may assume that none of the $v_{w_{i,j}}$ is ever elicited before any of the v_{x_i} or any of the v_{-x_i}. Finally, we observe that for each voter (besides the ones that approve only p) there is precisely one x_j that it (with certainty) does not approve; now, because there are only q elicitations left after eliciting the voters that approve only p, and each of the x_j needs to not be approved by at least one more elicited voter in order for us to be able to terminate, it follows that we will certainly not be able to terminate if (after eliciting the voters that approve only p) we elicit more than one voter that does not approve a given x_j. Thus we may assume that the contingency plan never does this.

We now show by induction on j that for $j>0$, the $|W|+q+j$th elicited voter must be either v_{x_j} or v_{-x_j}, unless the previously elicited voters have already eliminated all the elements of Cl. Suppose this is true for all $j<J$; we now show it true for all $j \leq J$.[12] We will suppose the contrary and derive

[12]Note that the base case with $J=0$ is vacuously true.

a contradiction. So, we suppose there exists a node in the elicitation tree at depth $|W|+q+J$ where not all the elements of Cl have been eliminated yet, but the voter elicited here is neither v_{x_J} nor v_{-x_J}. We know that on the path from the root to this node, the $|W|+q+i$th ($1 \leq i < J$) elicited voter was either v_{x_i} or v_{-x_i} (the induction assumption applies because obviously, at all of the nodes on the path to this node there were noneliminated candidates in Cl as well). We also know that none of these voters cast their third possible vote, because this would have eliminated all the elements of Cl. It follows that the elements of $\bigcup_{1 \leq j < J} \{x_j\} \cup W_j$ have already been eliminated. We first present an alternative contingency plan from this node onwards, which does elicit v_{x_J} at this node; then we show this alternative contingency plan performs better, contradicting the optimality of the original contingency plan. The alternative plan is as follows. As long as none of the voters have turned out to cast their third possible vote, we choose v_{x_j} as the $|W|+q+j$th elicited voter (for $j \geq J$). Otherwise, if v_{x_i} turned out to cast its third possible vote, from here on we choose $v_{w_{i,j}}$ as the $|W|+q+j$th elicited voter (for $j > i$). It is easy to see that if one of the v_{x_j} (for $j \geq J$) casts its third possible vote, this alternative plan will indeed allow us to terminate after k votes, and the probability that this occurs (starting at the given node) is $1 - (\frac{1}{2})^{q-J+1}$.

Now we proceed to analyze the original, supposedly optimal plan that elicits a voter other than v_{x_J} and v_{-x_J} at the given node. The voter elicited here cannot be one of those approving only p, since we could assume these voters are all elicited in the first $|W|+q$ elicitations. The voter elicited can also not be one of the $v_{w_{i,j}}$, for if it is, all the voters elicited after this will also be from the $v_{w_{i,j}}$ (because we could assume none of the $v_{w_{i,j}}$ are ever elicited before any of the v_{x_i} or any of the v_{-x_i}), the noneliminated elements of Cl will thus never be eliminated, and we will certainly not be able to terminate after k elicitations. This conflicts with the optimality of the plan because the alternative contingency plan we presented did allow us to terminate after k elicitations with nonzero probability. Hence, the elicited voter must be some v_{x_i} or v_{-x_i} with $i \neq J$. But it cannot be the case that $i < J$, because in this case we have already elicited one of v_{x_i} and v_{-x_i} (by the induction assumption); and eliciting the other as well would conflict with the assumption we could make that the contingency plan never elicits more than one voter that does not approve a given x_j (after eliciting the voters that approve only p). So $i > J$. Now consider the candidate $w_{J,i}$, which we still need to eliminate. We cannot elicit $v_{w_{J,i}}$ to eliminate this candidate, because eliciting both this voter and one of v_{x_i} and v_{-x_i} conflicts with the assumption we could make that the contingency plan never elicits more than one voter that does not approve a given x_j. It follows that the only voters that we might elicit to eliminate $w_{J,i}$ are v_{x_J} and v_{-x_J} (but, by the same assumption again, at most one of them). Because with probability $\frac{1}{2}$ such a voter will cast its third possible vote, which will not eliminate $w_{J,i}$, we will terminate with probability at most $\frac{1}{2}$. But the alternative strategy terminates with probability at least $1 - (\frac{1}{2})^{q-J+1} \geq \frac{3}{4}$ (the inequality follows from the fact that $q \geq i > J$), and thus the original strategy is not optimal. (Contradiction.)

It follows that we can make the part of the contingency plan where no voter has cast its third possible vote correspond to a contingency plan for the SSAT instance: when voter v_{x_i} (v_{-x_i}) is elicited, x_i is set to *true* (*false*), and when this voter responds to eliminate $sat(y_i)$ ($sat(-y_i)$), y_i is set to *true* (*false*). We claim this contingency plan is in fact a solution to the SSAT instance. For with probability $(\frac{1}{2})^q$, no elicited voter will cast its third possible vote; and, because by assumption, we can eventually terminate with probability at least $1 - (\frac{1}{2})^{q+1}$, it must be the case that given that this happens, we can still terminate with probability at least $\frac{1}{2}$. But, given that no elicited voter casts its third possible vote, we can terminate elicitation if and only if all elements of Cl are eliminated (which in turn happens if and only if in the corresponding event in the SSAT instance, all the clauses are satisfied). So there exists a solution to the SSAT instance. ■

Although $\mathcal{PSPACE}$-hardness is a stronger claim than $\mathcal{NP}$-hardness, 1) both are worst-case measures, 2) with the current state of knowledge in computational complexity, both seem intractable, and 3) $\mathcal{PSPACE}$-hardness implies nothing more about inapproximability than $\mathcal{NP}$-hardness. On the other hand, we believe that showing that effective elicitation is hard even with perfect suspicions about how the voters will vote is very significant. Thus, we prefer the $\mathcal{NP}$-hardness results for the case where the elicitor has perfect suspicions about the votes, to $\mathcal{PSPACE}$-hardness results for the more general case of imperfect suspicions.