HOW HARD IS IT TO CONTROL AN ELECTION?

John J. Bartholdi, III
School of Industrial and Systems Engineering
Georgia Institute of Technology, Atlanta, GA 30332, USA

Craig A. Tovey
School of Industrial and Systems Engineering
and College of Computing
Georgia Institute of Technology, Atlanta, GA 30332, USA

Michael A. Trick
Graduate School of Industrial Administration
Carnegie Mellon University, Pittsburgh, PA 15213, USA

December 1987; revised October 1990, April 4, 2003

Abstract

Some voting schemes that are in principle susceptible to control are nevertheless resistant in practice due to excessive computational costs; others are vulnerable. We illustrate this in detail for plurality voting and for Condorcet voting.

1 Introduction

Some voting procedures can be controlled by the authority conducting the election (hereinafter, the *chairman*) to achieve strategic results. For example, it might be possible to influence the outcome of an election by specifying the sequence in which alternatives will be considered, or by specifying the composition of subcommittees that nominate candidates. We study whether the chairman can easily or only with great difficulty determine how to control a specific election, and conclude that voting procedures can differ significantly in the effort

required to control them. This suggests that some voting procedures can be inherently resistant to abuse, while others are vulnerable. We base this distinction on a measure that is new to voting theory—computational complexity.

Let V be the set of voters, each of whom is identified with his preference order, which is assumed to be a transitive, complete, and strict order on the set C of candidates. We further assume that the chairman knows the preferences of every voter, and knows that they will vote sincerely. We emphasize that these assumptions are not intended to be realistic, but rather to be conservative: We shall show that even when the chairman has such strong, and possibly unrealistic, advantages, some types of manipulation can nevertheless be difficult.

Assuming the chairman knows beforehand the voting scheme under which the election will be held, can he fix some procedural matter so that his favorite candidate c, otherwise a non-winner, will become the unique winner of the election? We explore this canonical question for two types of voting procedures: Plurality voting and "Condorcet voting". Plurality voting selects as winner that candidate who has more first place votes than any other; for our purposes the Plurality winner is undefined in case of ties. By "Condorcet voting" we mean any procedure that always elects that candidate who would defeat any other in a pairwise election; if no such candidate exists then the Condorcet winner is undefined. In practice both Plurality voting and Condorcet voting are extended by some tie-breaking procedure to avoid undefined outcomes. We ignore tie-breaking rules, however, and restrict ourselves to the "pure" form of the voting scheme. Consequently our results are more general since they hold independently of which particular tie-breaking scheme might be adopted.

Plurality voting and Condorcet voting as we have defined them can be thought of as social choice functions; that is, each is a function E that, where defined, maps each (C, V) to a subset of C (Plott, 1976). In our model it is the goal of the chairman to ensure that $E(C, V) = \{c\}$, so that his favorite candidate is the unique winner.

2 Computational complexity

We briefly sketch some of the issues and techniques of computational complexity. The interested reader should consult Garey and Johnson (1979) for more detail.

The theory of computational complexity is concerned with how much work is required to solve well-defined computational problems. For complexity analysis, a problem is a class of instances sharing a common form. This form is precisely described in terms of the data that make up an instance and the question to be answered.

The amount of work required to solve an instance of the problem is measured as a function of the size of the instance (which is $O(|C||V|)$ for an election—the space required to list the preferences of the voters). The key distinction to be made is the following. If a solution method requires time that grows as a *polynomial* in the size of the instance, then it is considered fast; if it requires time that grows as an *exponential* function, then it is considered slow. In fact, because exponential functions grow so much more quickly than polynomials, an exponential-time algorithm is impractical for all but small or specially structured instances of a problem. This distinction has been widely observed to be consistent with the judgment of practical experience.

Two classes of problems are of particular interest. Class NP consists of all "yes/no" questions for which a potential solution can be verified in polynomial time (even though it might take exponential time to have computed the solution). Class P consists of those problems of NP for which a solution can always be computed in polynomial time.

The key technique we employ is "polynomial transformation". Suppose we can recast any instance of problem p as an instance of problem q. If this transformation is fast then the task of solving problem p *polynomially transforms* to that of solving q. We could then conclude that if problem q can be solved quickly, so can p. (Hereafter, since we are only concerned with polynomial transformations, we will generally omit the word "polynomial".)

Some pairs of problems have the property that each transforms to the other;

such pairs are termed *polynomially equivalent* because a fast (that is, polynomial-time) method for solving one of the pair implies a fast method for the other. The *NP-complete* problems are a famous subclass of NP, all of whose members are polynomially equivalent and for which no dependably fast solution method is known. These problems include such notoriously difficult problems as integer programming, the traveling salesman problem, determining whether a voter has non-zero "Shapley-Shubik voting power", and hundreds of others from a variety of fields. It is widely accepted that, unless specially-structured or "small", NP-complete problems are both theoretically and practically difficult to solve. We will show that some problems of agenda control are also NP-complete, and so as difficult as all these famous hard problems.

It is important to note that NP-completeness is an asymptotic measure of complexity, and so should be interpreted as an indication of how quickly the work-to-solve increases as a function of problem size. NP-completeness does not say whether a specific instance is difficult. In fact, any problem, the size of whose instances is bounded in advance, is technically easy: solution by complete enumeration requires work that is bounded by a (possibly very large) constant. Thus it is more informative to interpret NP-completeness as indicating that any solution method will likely require work that increases exponentially in some aspect of the size of the data. Frequently this critical aspect is the number of candidates in the election.

We say that a voting scheme is *immune* to control if it is never possible for the chairman to change c from a non-winner to a unique winner by manipulating procedural matters; otherwise the scheme is *susceptible* to control. For a voting scheme that is susceptible to control, if it is always computationally easy to recognize opportunities for control, then we say the voting scheme is *vulnerable*; if it is hard (that is, NP-complete) to recognize opportunities for control, then the voting scheme is *resistant* to control. This view explicitly accounts for the limited computational abilities of the participants. This can be seen as another elaboration of the idea of "bounded rationality" that has enriched so many other economic and social models (Simon, 1972).

The classification of voting procedures by their computational complexity has also appeared in Bartholdi, Narasimhan, and Tovey (1990), Bartholdi, Tovey, and Trick (1989a), and Bartholdi, Tovey, and Trick (1989b). Both this paper and Bartholdi, Tovey, and Trick (1989b) have been influenced by Nurmi (1984, p. 255), who suggested "... constructing a hierarchy reflecting the difficulty of benefiting from strategic behavior".

3 Procedural control

The types of control we formalize can involve adding to, deleting from, or partitioning either the set of candidates or the set of voters. Most typical voting schemes are in principle susceptible to these types of control. However, as we shall show, voting schemes can differ greatly in susceptibility: some are computationally vulnerable while others are computationally resistant to control. We illustrate these differences for Plurality voting and Condorcet voting.

A note about the proofs: Transformation arguments can be intricate and highly formal. Furthermore, since the polynomially-equivalent problem might not bear any obvious intuitive relation to the voting problem, the proof might establish computational difficulty without seeming to explain its source. In this sense, the conclusion is more important than the argument. Accordingly, we consign all but one of the proofs to an appendix.

3.1 Agenda Control of Candidates

3.1.1 Adding candidates

One type of control is for the chairman to add "spoiler" candidates to the slate in hopes of diluting the support of those who might otherwise defeat c.

Some voting procedures are immune to control by adding candidates because they satisfy the Weak Axiom of Revealed Preference (WARP), which requires that the winner among a set of candidates be the winner among every subset of candidates to which he belongs (Plott, 1976). (This is also known as prop-

erty α (Kelly, 1988)). Thus, for a voting procedure satisfying WARP, if c can be made the winner in an augmented election, then c must have been a winner in the original election. For voting procedures that are susceptible to this kind of control, we can still distinguish between computational resistance and computational vulnerability.

Plurality voting does not satisfy WARP (Nurmi, 1983). However, we show that it can be computationally difficult to recognize when a plurality election fails to satisfy WARP, so it is difficult for a chairman to take advantage of this fact.

We follow the conventions of computational complexity in which problems are formalized by giving the data required for an instance and the question that must be answered. For technical reasons, this question is phrased as a "yes or no" question.

Control by Adding Candidates
GIVEN: A set C of qualified candidates and a distinguished candidate $c \in C$, a set B of possible spoiler candidates, and a set V of voters with preferences over $C \cup B$.
QUESTION: Is there a choice of candidates from B whose entry into the election would assure victory for c?

To a certain extent the exact formalization of a problem is a matter of taste. For example, while we have formalized this question to be whether there is some subset of B whose entry would result in the election of c, we could equally well have formalized it to be whether there are K or fewer candidates to be added from B. The first formalization emphasizes recognizing whether WARP holds. The second formalization corresponds to the minimization problem of finding the smallest number of candidates the chairman must add to elect c. This emphasizes agenda control, for the chairman would need to make the agenda changes as innocuous as possible. Context determines which formalization is the more natural. It does not much matter for the problems we discuss, since both versions are of the same complexity.

We now state and prove the result. For clarity, we give full details and also include an example of the transformation; subsequent proofs are briefer and relegated to an appendix.

Theorem 1. *Plurality voting is computationally resistant to control by adding candidates.*

Proof. We show that the problem is NP-hard by reformulating the following NP-complete problem as a control problem:

Hitting Set (Garey and Johnson, 1979, p. 222)
GIVEN: Finite set $B = \{b_1, \ldots, b_m\}$, a collection $S = \{S_1, \ldots, S_n\}$ of subsets of B and a positive integer $K \leq m$.
QUESTION: Is there a subset $B' \subseteq B$ with $|B'| \leq K$ such that B' contains at least one element from each subset in S?

For example, we might have $n = 9, K = 2, m = 5$, and, denoting b_i as i for convenience,

$$S = \{(1,3,5), (1,2,4), (3,4,5), (2,3,4), (1,4,5), (2,3,5), (1,2,3), (2,4,5), (1,2,5)\}$$

For this instance the answer is "yes"; the solution happens to be unique: $B' = \{b_2, b_5\}$. Intuitively, this problem is hard because one has to check all the possible K-subsets to see which if any works.

Now we show how an instance of Hitting Set can be transformed to an instance of Control by Adding Candidates to a Plurality Election. From an arbitrary instance of Hitting Set, contrive an election with qualified candidates c, c', and d, and unqualified candidates corresponding to the b_i. The idea of the proof is to create an electorate whose preferences are such that c' will win the election and c will finish second if no control is exercised by the chairman, who wishes c to win. Furthermore, the voter preferences are such that adding new candidates help c gain votes relative to c', but also unavoidably helps d gain on c. If the chairman adds too many candidates d will beat c, so he must find a set of candidates to add that is small enough to prevent d winning but large enough to enable c to overtake c'.

The voter preferences of the contrived election are as follows. Let there be $2n - m$ voters who prefer c to all other candidates; let there be $2n - m - 1$ voters who prefer c' to all other candidates; let there be $2n - K - 1$ voters who most prefer d. Let there be n voters corresponding to the elements of S, where the voter S_j most prefers (in any order) those candidates $b_i \in S_j$; next prefers c', followed (in any order) by the other candidates. (Notice that the voter's preferences lower than the first qualified one (that is, c, c', d) are irrelevant.) Finally, for $i = 1, \ldots, m$ let there be 1 voter who most prefers b_i, and next prefers c; and one voter who most prefers b_i, and next prefers c'. We call these last $2m$ voters "single voters."

In an election among the three qualified candidates, c' must win, with $3n-1$ votes to only $2n$ votes for c and $2n-K-1$ for d. However, introducing candidate b_i into the election will cause the voters corresponding to S_j containing b_i to switch their votes from c' to b_i. It will also cause both c and c' (but not d) to lose one vote from the single voters. Now if there exists a hitting set B' of cardinality $|B'| \leq K$, then introducing the b_i of the hitting set as candidates will induce each of the n voters S_j to switch his vote from c' to one of the $b_i \in S_j$. Also, c and c' will each lose $|B'| \leq K$ votes due to single voter defections to members of B'. This leaves c with $2n-|B'|$ votes; c' with $3n-1-|B'|-n = 2n-|B'|-1$ votes; d still with $2n - K - 1 \leq 2n - |B'| - 1$ votes; and no b_i in B' with more than $n + 2$ votes. Candidate c is the winner.

We have established that if the answer to the Hitting Set instance is "yes", then the answer to the agenda control instance is also "yes". To prove the converse, assume that introducing the candidates of B' into the election will cause c to win. Then at least n more voters must have defected from c' than from c. But single voters defect from c and c' in equal numbers, so these n could only have been the voters corresponding to the S_j. Furthermore, each S_j must have found a new candidate corresponding to one of the $b_i \in S_j$. Thus the new candidates B' must form a hitting set with respect to the S_j.

If c will win, c must defeat d as well as c'. Candidate d will get $2n - K - 1$ votes, while c will get $2n - |B'|$ votes. Therefore $|B'| \leq K$. Thus the

transformation is *correct* for the answer to the agenda control instance is "yes" if-and-only-if the answer to the instance of Hitting Set is "yes".

To complete the proof, observe that the transformation requires only polynomial time. □

As an illustration of the transformation, the example of Hitting Set above is transformed to: $C = \{c, c', d\}$, $C' = \{b_1, b_2, b_3, b_4, b_5\}$; $V = \{13$ of $(c, \ldots)$; 12 of $(c', \ldots)$; 16 of $(d, \ldots)$; the S_j voters $(b_1, b_3, b_5, c', \ldots)$, $(b_1, b_2, b_4, c', \ldots)$, $\ldots, (b_1, b_2, b_5, c', \ldots)\}$. The only way to make c win is to qualify b_2 and b_5. Thus one must solve the embedded hitting set problem to determine which candidates to qualify.

Unlike plurality elections, Condorcet elections are immune to control by adding candidates for the obvious reason that if c can be defeated by some other candidate, this fact is unchanged when additional candidates are added to the election. Adding candidates might change the Condorcet winner to one of the new candidates, but can never make c Condorcet.

Theorem 2. *Condorcet voting is immune to control by adding candidates.*

3.1.2 Deleting candidates

Another way in which the chairman might try to influence the election is by disqualifying some candidates, whose supporters might then rally to c.

Control by Deleting Candidates
GIVEN: A set C of candidates, a distinguished candidate $c \in C$, a set V of voters, and a positive integer $K \leq |C|$.
QUESTION: Are there K or fewer candidates whose disqualification would assure the election of c?

In a plurality election it can be hard to know which candidates to disqualify. The supporters of a disqualified candidate will switch to other candidates, but it can be difficult to judge the overall effect of the switching.

Theorem 3. *Plurality voting is computationally resistant to control by deleting candidates.*

On the other hand, it is easy to judge the effect of deleting candidates from a Condorcet election:

Theorem 4. *Condorcet voting is computationally vulnerable to control by deleting candidates.*

3.1.3 Partitioning candidates

Consider an election that takes place in two stages, based on a partition of the candidates into subsets C_1 and C_2 so that $C_1 \cup C_2 = C$ and $C_1 \cap C_2 = \emptyset$. First the entire electorate votes on the candidates from C_1; then the winner of that election goes on to face the candidates of C_2. Can the chairman influence the election by his choice of the partition C_1, C_2? We formalize this question as follows.

Control by Partition of Candidates
GIVEN: A set C of candidates, a distinguished candidate $c \in C$, and a set V of voters.
QUESTION: Is there a partition of the candidates into C_1, C_2 so that c is the unique winner in sequential elections? (That is, so that $\{c\} = E(V, E(V, C_1) \cup C_2)$.)

A voting scheme satisfies *path-independence* if, for any V, C, $E(V, C) = E(V, E(V, C_1) \cup C_2)$ for any C_1, C_2 for which $C_1 \cup C_2 = C$ and $C_1 \cap C_2 = \emptyset$ (Plott, 1973; Plott, 1976). Control by Partition of Candidates exploits possible path-*dependence* in an election, and can thus be interpreted as asking whether a particular election violates path-independence.

Neither plurality voting nor Condorcet voting are path-independent (Nurmi, 1983). However, to recognize violations of path-independence can be computationally difficult in a plurality election, but is always easy in a Condorcet election:

Theorem 5. *Plurality voting is computationally resistant to control by partition of candidates.*

Theorem 6. *Condorcet voting is computationally vulnerable to control by partition of candidates.*

A special case of partitioning candidates occurs in sequential pairwise voting, in which the current incumbent is matched against a sequence of individual challengers in successive two-candidate elections, with winners decided by majority vote. (Both plurality and Condorcet voting are identical to majority voting when there are only two candidates.) The ultimate winner is the candidate who emerges victorious from the final election.

It is well known that the ultimate winner of such a process can sometimes be determined entirely by the sequence of comparisons (see, for example, Banks, 1985; Harary and Moser, 1966; MIller, 1980). In fact, the chairman can determine a sequence resulting in victory for c, or conclude that none exists, within a polynomial number of computational steps. To see this, construct the tournament G of all pairwise comparisons between candidates, where an arc is directed from candidate i to candidate j if i defeats j. (This requires $O(|C|^2|V|)$ steps.) Now use breadth-first search (Sedgewick, 1988) to determine whether there exists a tree rooted at c that spans all the candidates. (This requires $O(|C|)$ steps.) If so, then c will emerge victorious from the sequence in which candidates appear in non-increasing order of their distance from c in G. On the other hand, if no tree rooted at c is spanning, then some candidate not in the maximal tree rooted at c must ultimately win the election.

An alternate way to utilize a partition of the candidates is for the electorate to vote separately on the candidates from C_1 and C_2; then the winners of the two elections face each other in a "run-off" election. For such a structure we formalize the chairman's control problem as follows.

Control by Run-Off Partition of Candidates
GIVEN: A set C of candidates, a distinguished candidate $c \in C$, and a set V of voters.

QUESTION: Is there a partition of the candidates into C_1, C_2 so that c is the unique winner in runoff elections? (That is, so that $\{c\} = E(V, E(V, C_1) \cup E(V, C_2))$.)

Neither plurality voting nor Condorcet voting is immune to this type of agenda control. As with candidate partition, plurality voting is computationally resistant while Condorcet voting is computationally vulnerable.

Theorem 7. *Plurality voting is computationally resistant to control by run-off partition of candidates.*

Theorem 8. *Condorcet voting is computationally vulnerable to control by run-off partition of candidates.*

3.2 Manipulating voters

3.2.1 Adding voters

Another strategy to control an election is to register new voters to aid the cause of c. Since we assume that all voters remain sincere in their preferences, the chairman must decide which of a fixed set of additional voters he should encourage to vote. Presumably the chairman would prefer to accomplish his goal by registering a small number of additional voters. Accordingly, we formalize his problem as one of minimizing the number of new voters that ensure victory for c.

Control by Adding Voters
GIVEN: A set C of candidates and a distinguished candidate $c \in C$; a set V of registered voters and an additional set V' of voters who are unregistered but could still register in time for the election; a positive integer $K \leq |V'|$.
QUESTION: Are there K voters from V' whose registration would assure the election of c?

This is different from the problem of strategic voting, which asks whether there is a preference order that can be assumed by a voter to ensure victory

for c (Bartholdi, Tovey, and Trick, 1989b). In strategic voting the voter is free to profess whatever preferences he wishes, while for control by adding voters, preferences can be added only from among those of a fixed known set.

It is easy to decide which voters to add to a plurality election because plurality voting incorporates only information about each voter's favorite candidate. On the other hand it can be hard to decide which voters to add to a Condorcet election since the entire preference order of each voter affects the election.

Theorem 9. *Plurality voting is computationally vulnerable to control by adding voters.*

Theorem 10. *Condorcet voting is computationally resistant to control by adding voters.*

3.2.2 Deleting voters

A dual strategy is to disenfranchise voters to the detriment of any candidate who would otherwise defeat c.

Control by Deleting Voters
GIVEN: A set C of candidates and a distinguished candidate $c \in C$; a set V of voters; and a positive integer $K \leq |V|$.
QUESTION: Are there K or fewer voters whose disenfranchisement would assure the election of c?

Again, it is easy to identify which voters to delete from a plurality election because the voting scheme considers only first-place votes, but it is hard to tell which voters to delete from a Condorcet election because each voter affects every pairwise contest.

Theorem 11. *Plurality voting is computationally vulnerable to control by deleting voters.*

Theorem 12. *Condorcet voting is computationally resistant to control by deleting voters.*

3.2.3 Partitioning voters

Consider an election in which the voters are divided into two "committees", each of which holds an independent election to select a nominee; then the entire electorate votes on the two nominees to select a final winner. In such an election the chairman might be able to influence the final outcome by his selection of the committees:

Control by Partition of Voters
GIVEN: A set C of candidates and a distinguished candidate $c \in C$; a set V of voters.
QUESTION: Is there a partition of the voters into V_1, V_2 so that hierarchical elections assure the victory of c? (That is, so that $\{c\} = E(V, E(V_1, C) \cup E(V_2, C)))$

A voting scheme is immune to this type of control if it possesses what might be called *strong consistency*: for any partition of V into V_1 and V_2, $E(V, E(V_1, C) \cup E(V_2, C)) = E(V, C)$, so that the winner of the election is independent of the nominating subcommittees. As suggested by the name, strong consistency is an extension of a more commonly-studied property, consistency, which requires that, for any partition of V into V_1 and V_2, $E(V_1, C) \cap E(V_2, C) \subseteq E(V, C)$; that is, any candidate nominated by both committees must be a winner. Both plurality voting and Condorcet voting are consistent. However, neither is strongly consistent, and so both can in principle be controlled by choosing the subcommittees.

Theorem 13. *Plurality voting is computationally vulnerable to control by partition of voters.*

This can be interpreted as establishing that it is easy to check for violations of strong consistency in a plurality election. In contrast, the following theorem shows that it can be hard to recognize such violations in Condorcet voting.

Theorem 14. *Condorcet voting is computationally resistant to control by partition of voters.*

Control by ...	Plurality	Condorcet
adding candidates	resistant	immune
deleting candidates	resistant	vulnerable
partitioning candidates	resistant	vulnerable
adding voters	vulnerable	resistant
deleting voters	vulnerable	resistant
partitioning voters	vulnerable	resistant

Table 1: The computational difficulty of control by the chairman.

4 Conclusions

Table 1 summarizes the difficulties of controlling plurality voting and Condorcet voting. We chose these two voting procedures to illustrate how resistance the susceptibility to control can differ significantly among procedures.

In general, plurality voting resists control of candidates, while Condorcet voting resists control of voters. We can explain this intuitively.

Plurality voting, voter control For a fixed slate of candidates plurality voting is simple to control because each voter is sure to vote for exactly one candidate. If we consider adding two voters, v_1 and v_2, the effect of v_1 (a single vote for his candidate) is the same whether v_2 is added or not. The control issues involve simply counting "good" and "bad" voters to determine victory and so tend to be easy.

Plurality voting, candidate control When the slate of candidates can be changed, the choice of a voter depends on exactly which candidates are added. If we consider adding two candidates, c_1 and c_2, the effect of c_1 may be different if c_2 is added also. Voters (or candidates) are no longer simply "good" or "bad": instead their effect depends on other choices. Because so many interdependencies must be accounted for, the control problems tend to be hard.

Condorcet voting, candidate control When the set of voters is fixed, the tournament graph of all pairwise election outcomes is fixed. All that changes is the subset of vertices under consideration. If we consider adding two candidates (i.e. vertices) c_1 and c_2, the edges added to the graph for c_1 are the same whether or not c_2 is added (except for the single edge to c_1). Therefore adding candidates does not have cross-effects, and thus the control problems tend to be easy.

Condorcet voting, voter control If the set of voters can be changed, complicated interactions can occur. A single voter can tip the balance in several pairwise elections; another voter can help tip the balance of the same elections in a different pattern. The effects of multiple voters will partially cancel and partially reinforce each other in complicated ways, and so the control problems tend to be hard.

5 Practical implications

Computational complexity can help make finer distinctions than have heretofore been made concerning the susceptibility of voting schemes to control. Previously, the logical *possibility* of control has been determined; our goal has been to measure its *practicality* and one measure of practicality is computational requirements. We hope to identify difficulty that is intrinsic to the voting scheme and so provides worst-case protection.

Our results suggest that some voting schemes might resist control even by a chairman armed with a large computer and perfect information. This suggestion is based on the fact that NP-hard problems are both theoretically and practically difficult to solve. However, we cannot conclude that "typical" elections will be hard to control, because we do not know what a typical election is. Also, while NP-completeness is taken as strong circumstantial evidence of inherent and pervasive difficulty, strictly speaking it does not prove that. The difficulty is that NP-completeness is an asymptotic worst-case measure of difficulty, and so,

strictly speaking, we can conclude only that, for "large enough" elections, some instances can be difficult to control. This deserves further discussion. First, what is "large enough"? This is an empirical question that must be resolved by computational test. One apparent implication of NP-completeness is that any solution procedure must require time that increases exponentially in the size of the problem (Garey and Johnson, 1979, p. 14, 121–122). Thus, because of the explosive growth of exponentiality, the time budget of any computing device, however fast the device, must eventually be overwhelmed; but exactly when this occurs depends on both the problem and the device. For example, if the computing device is a person with pencil and paper, then control of even a small election is likely to require excessive computation time, so our results can have practical implications even for committee elections.

The second concern with NP-completeness is that it is a worst-case measure of difficulty and so, strictly speaking, says nothing about the frequency with which hard instances will be encountered. Of course we would prefer the stronger guarantee that a "typical" election is difficult to control. Unfortunately, such a result seems beyond current complexity theory. Thus our work complements, but does not supersede, the usual approach to measuring susceptibility to control, which is to count the instances in which control is logically possible. Unfortunately, this approach has weaknesses too. For example, in the standard approach it is impossible to justify the distribution from which election instances are drawn. Frequently it is quite unrealistic and is chosen simply for convenience of analysis.

In any event, we can make some conclusions about the relative difficulty of control problems. First, because NP-complete problems apparently require exponential-time solution procedures, it is harder to manipulate voters than candidates in a Condorcet election. The conclusions are not so clear under a Plurality election: It is NP-complete to manipulate candidates, but there are generally many fewer candidates than voters. Therefore, even if the work to manipulate candidates increases exponentially in the number of voters and the work to manipulate voters increases only polynomially in the number of

voters, the relative efforts could be comparable. We can only say that if there are "sufficiently many" candidates in the Plurality election, manipulation of candidates will likely be impractical.

Despite the fact that NP-completeness is a worst-case measure of difficulty, it might still provide practical assurances of difficulty, if not guarantees. Indeed, throughout computer science, engineering, and operations research, NP-hardness is taken as strong theoretical evidence of the impracticality of solving large problems. It has continued to be the experience of many people in many disciplines that all but small or specially-structured NP-complete problems are inordinately time-consuming to solve. In fact, there is sufficient confidence in the inherent difficulty of NP-complete problems that cryptographic schemes have been based on it (Diffie and Hellman, 1976)—and all NP-complete problems are equally difficult, at least in a formal sense.

Even though NP-completeness cannot provide absolute guarantees of difficulty, it does have immediate practical implications for the behavior of the chairman. For example, when a problem is NP-complete, there is apparently no solution procedure that is significantly better than brute-force enumeration over a potentially very large set of possibilities. There can be no direct and fast method, no rule or recipe, to recognize opportunities for control; instead one is forced to sift through exponentially many possibilities (compare Theorems 1 and 2). Thus our results explain how a manipulative chairman is constrained to plan his strategy when the problem of control is NP-complete.

Finally, we suggest that computational complexity be one of the criteria by which voting schemes are routinely evaluated. How much effort is required to determine a winner? How much effort is required to control? We have shown that there can be qualitative differences in the answers to such questions; it might be that these qualitative differences are also practical differences. Such issues will become more important as our social choice processes are implemented increasingly often on computers.

Acknowledgements

We thank Steven Hackman, Walter Mueller, David Nachman, Reuben Saposnik, John Vande Vate, the editor, and two anonymous referees for helpful comments.

John Bartholdi was supported in part by a Presidential Young Investigator Award from the National Science Foundation (ECS-8351313), and by the Office of Naval Research (N00014-85-K-0147); Craig Tovey was supported in part by a Presidential Young Investigator Award from the National Science Foundation (ECS-8451032); Michael Trick was supported in part by the Office of Naval Research (N00014-85-K-0147) and postdoctorate fellowships from NATO, The Institute for Mathematics and Its Applications, Minneapolis, MN, USA, and the Institut für Ökonometrie und Operations Research, Bonn, Federal Republic of Germany.

References

[1] Banks, J. S. 1985. Sophisticated voting outcomes and agenda control. *Social Choice and Welfare*, 1:295–306.

[2] Bartholdi, J. J. III, L. S. Narasimhan, and C. A. Tovey. 1990. Recognizing majority-rule equilibrium in spatial voting models. *Social Choice and Welfare*, to appear.

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

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

[5] Diffie, W. and M. Hellman. 1976. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22:644–654.

[6] Garey, M. and D. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, San Francisco.

[7] Harary, F. and L. Moser. 1966. The theory of round robin tournaments. *American Mathematical Monthly*, 73:231–246.

[8] Kelly, J. S. 1988. *Social Choice Theory*, Springer–Verlag, New York.

[9] Miller, N. 1980. A new solution set for tournaments and majority voting: further graph–theoretic approaches to the theory of voting. *American Journal of Political Science*, 24:68–96.

[10] Nurmi, H. 1983. Voting procedures. *British Journal of Political Science* 13:159–186.

[11] Nurmi, H. 1984. On the strategic properties of some modern methods of group decision making. *Behavioral Science* 29:248–257.

[12] Plott, C. R. 1973. Path independence, rationality, and social choice. *Econometrica* 41(6):1075–1091.

[13] Plott, C. R. 1976. Axiomatic social choice theory: an overview and interpretation. *American Journal of Political Science* 20:511–596.

[14] Sedgewick, R. 1988. *Algorithms* (Second Edition), Addison-Wesley Publishing Company.

[15] Simon, H. 1972. Theories of bounded rationality. In *Decision and Organization*, edited by C. McGuire and R. Radner, North-Holland Publishers, Amsterdam.

A Proofs

A.1 Manipulating candidates

Theorem 2. *Condorcet voting is invulnerable to control by adding candidates.*

Proof. If c is currently a Condorcet winner, then he can be maintained so by adding no additional candidates. If c is not currently a Condorcet winner, he cannot be made so by adding any candidates. □

Theorem 3. *Plurality voting is computationally resistant to control by deleting candidates.*

Proof. We show that the problem is NP-hard by transformation from the following problem that is known to be NP-hard (Garey and Johnson, 1979, p. 221).

Exact Cover by 3-Sets (X3C)
GIVEN: A set $B = \{b_1, \ldots, b_m\}$, where m is evenly divisible by 3, and a collection $S = \{S_1, S_2, \ldots, S_n\}$ of 3-element subsets of B.
QUESTION: Does S contain an exact cover for B—that is, a subcollection $S' \subseteq S$ such that every element of B occurs in exactly one member of S'?

Let b_i^1, b_i^2, and b_i^3 denote the elements of S_i. For an arbitrary instance of X3C, create an election with the following candidates:

c		: the intended winner;
w		: the current winner;
s_i,	$i = 1, \ldots, n$	: corresponding to the S_i;
b_j,	$j = 1, \ldots, m$	: corresponding to the elements of B;
a_k,	$k = 1, \ldots, m/3$	: additional candidates

There are several groups of voters, as follows (where, in the preference orders, entries below the last one specified are in arbitrary order). First, for each i there is 1 voter with preferences $s_i > c > a_1 > \ldots > a_{m/3} > \ldots$. These first voters are important because each ranks c highly. Each of the remaining voters ranks c below at least $(m/3) + 1$ other candidates.

Next, for each i there is 1 voter with preferences $s_i > b_i^1 > a_1 > \ldots > a_{m/3} > \ldots$, 1 voter with preferences $s_i > b_i^2 > a_1 > \ldots > a_{m/3} > \ldots$, and 1 voter with preferences $s_i > b_i^3 > a_1 > \ldots > a_{m/3} > \ldots$.

The remaining voters form blocks of votes, the largest of which consists of $(m/3) - 1$ voters all with the preferences $w > a_1 > \ldots > a_{m/3} > \ldots$. In addition, for every j there are $(m/3) - 2$ voters all with the preferences $b_j > a_1 > \ldots > a_{m/3} > \ldots$.

We claim that the instance of X3C has a solution if and only if one can remove no more than $m/3$ candidates from this election so that c wins.

Suppose there exists an exact 3-cover; then delete those candidates s_i corresponding to the S_i in the cover. Then, since an exact 3-cover has cardinality exactly $m/3$, c receives $m/3$ votes; each b_j gains 1 vote for a total of $(m/3) - 1$ votes; votes for w are unchanged at $(m/3) - 1$. All other candidates receive less than $(m/3) - 1$ votes, so c is the winner.

Now suppose that there exists a subset of no more than $m/3$ candidates whose deletion leaves c the victor. In this transformed election, c can hope to get votes only from the voters $s_i > c > \ldots > a_{m/3}$, because the position of c in all other preferences is lower than $m/3$. Thus all deletions must be of candidates s_i, and c can get no more than $m/3$ votes. Furthermore, c must receive at least $m/3$ votes since the block of $(m/3) - 1$ voters who favor w will always vote as a group for some candidate other than c. Therefore, c must receive exactly $m/3$ votes, and this can be accomplished only by deleting exactly $m/3$ of the candidates s_i. But if the corresponding S_i do not form a cover, then some b_j receives 2 additional votes and c does not defeat him. Thus the corresponding S_i must comprise an exact 3-cover. □

Theorem 4. *Condorcet voting is computationally vulnerable to control by deleting candidates.*

Proof. If the number of candidates who defeat c in pairwise contests is less than or equal to K, then these can be deleted to make c a Condorcet winner; otherwise c cannot be made a Condorcet winner.

The number of candidates who defeat c in pairwise contests can be determined in $O(|C|^2|V|)$ time. □

Theorem 5. *Plurality voting is computationally resistant to control by partition of candidates.*

The proof for this theorem is a modification of the proof that plurality voting is computationally resistant to control by Runoff Partition of the candidates. Hence, the proof for this theorem follows the proof of theorem 7.

Theorem 6. *Condorcet voting is computationally vulnerable to control by partition of candidates*

Proof. If c is a Condorcet winner against the set of all candidates, then any partition will do. Notice that if a candidate $d \in C_1$ wins, that is $d \in E(V, C_2 \cup E(V, C_1))$ then d must be a Condorcet winner against the set of all candidates. Therefore, if c is not such a Condorcet winner, c can only win if $c \in C_2$. This can happen only if, for some $y \in C$, c beats y in a pairwise election, and for all other candidates x, either c beats x or y beats x (or both). For each y, these properties are easy to check. If such a y is found, we put y and all candidates y defeats into C_1; c and all remaining candidates (which by the properties are defeated by c) are put into C_2. □

Theorem 7. *Plurality voting is computationally resistant to control by runoff partition of candidates*

Proof. We show this problem is NP-hard by transformation from the problem of Manipulation by Deleting Candidates for Plurality, shown to be NP-hard in Theorem 3.

Given an instance of Control by Deleting Candidates, with voters V, candidates C, goal winner c', and integer k' (the number that can be deleted), we construct an instance of Manipulation by Runoff Partition as follows:

Let $|V| = n$, $|C| = m$, and $k = k' + 1$. Assume (as we can from the transformation in Theorem 3) that $k' < n/2$ and m is even.

The candidates in our new election will be the original candidates, denoted $c_1, C_2, \ldots, c_m$, f, x, and a large (but polynomial) number of filler candidates

y_i, z_i, and w_i, each of which will appear near the top of a voter profile exactly once. Let $C^* = C - c'$.

There are three types of voters as follows (all unspecified candidates follow the last given candidate arbitrarily):

TYPE 1 voters: for each voter in the Deleting Candidates instance, create $m - k'$ voters with x most preferred, then the original preferences, then f, and then the rest of the candidates arbitrarily.

TYPE 2 voters: create $nk'/2$ pairs of voters with preferences y_i, z_i, C^*, c', f, x and y_i, z_i, C^*, c', x, f.

TYPE 3 voters: create n copies of the following types of voters (each row is a voter):

c_1, f, w_i, C^*, x, c'

c_2, f, w_i, C^*, x, c'

$\vdots$

c', f, w_i, C^*

$\vdots$

c_m, f, w_i, C^*, x, c'

The total number of voters is $2mn$.

We claim that c' wins under runoff partition of the candidates if and only if c' wins the original election with fewer than k candidates deleted.

($\Leftarrow$) Assume c' wins the original election with fewer than k candidates deleted. Place the deleted candidates in C_2 along with f, x, and all the z_i. Now c' wins in C_1 because it wins each subelection of Type 1 voters, each Type 2 voter votes for a unique y_i, and the Type 3 voters divide evenly among C. Candidate f wins in C_2, for only x can possibly beat him, but his score is lower. Finally, c' beats f in the runoff.

($\Rightarrow$) If c' is to win, the following sequence of claims must be true. (Assume c' is in C_1.)

Claim: f wins in C_2. Reason: c' beats only f in the final runoff.

Claim: x is in C_2. Reason: c' loses to x in C_1.

Claim: From any valid partition, we can create a valid new partition by placing all the w_i and z_i in C_1 and all the y_i in C_2. Reason: None of this movement can do anything other than give one vote to a w, y or z (who can not win with one vote) and/or reduce the vote of a losing candidate (that is, not c' or f). Hence the partition is still valid.

Claim: Fewer than k of the C candidates are in A_2. Reason: The score of x is at least $(m-k)n$. If k^* of the C candidates are in A_2 then the score of f score is $(m-k^*)n$. For f to beat x, it must be that $k^* < k$.

Claim: c' wins the original election after the candidates have been deleted. Reason: if c' does not uniquely win, then some other $c'' \in C$ does. But c'' otherwise dominates c' and so would win in C_1.

Taking the last two claims together implies that a valid partition has embedded within in it a solution to the original problem that deletes fewer than k candidates. □

Now, if we take the above construction and add new filler candidates v_i in second place for the Type 3 voters, then for c' to win by partition of candidates (in the sense of Theorem 5), x still must be eliminated in the original election. Only f can do this. When f is added to the remaining candidates, the v_i prevent f from winning the final election. Therefore, due to the argument above, c' will win only if the deletion problem is solved. This proves Theorem 5.

Theorem 8. *Condorcet voting is computationally vulnerable to control by runoff partition of candidates.*

Proof. We show how, in polynomial time, one can construct a partition (C_1, C_2) of the candidates that will result in victory for c, or else conclude that none exists. First, without loss of generality, place c in C_1. The algorithm to construct the partition tries each remaining candidate as the winner in C_2: Pick a candidate and place him in C_2 as the intended winner there. For the control to be successful, the winner from C_2 must be defeated by c in pairwise contest. Therefore, if the intended winner from C_2 can defeat c, replace him with an-

other, previously unconsidered candidate from $C-\{c\}$. Let the intended winner in C_2 be c'. Now test whether the candidates in $C-\{c,c'\}$ can be assigned to C_1 or C_2 while maintaining our intended winners. For each candidate, if he can be defeated by c, place him in C_1; otherwise, if he can be defeated by c', place him in C_2; if he defeats both c and c', then it is not possible for c to win in one half of the partition and c' to win in the other half. Choose a previously unconsidered candidate from $C-\{c\}$ and try again.

Since it takes $O(|C|)$ time to try whether a candidate can be the winner from C_2, and there are $O(|C|)$ candidates to be tried, the algorithm requires $O(|C|^2)$ time. □

A.2 Manipulating voters

Theorem 9. *Plurality voting is computationally vulnerable to control by adding voters.*

Proof. Evaluate the current election. If c is currently defeated by more than K votes, he cannot be made a plurality winner. If c is currently defeated by $k \leq K$ votes, simply scan the voters of V': If there are k voters who prefer c to all other candidates, adding those voters will make c a winner; otherwise c cannot be made a winner. This can be determined in $O(|V|+|V'|)$ time. □

Theorem 10. *Condorcet voting is computationally resistant to control by adding voters.*

Proof. We show that the problem is NP-hard by transformation from X3C.

For any instance of X3C create an election with candidates c and b_i ($i = 1,\ldots,m$). Let V consist of $(m/3)-3$ voters, all with the preference order $b_1 > \ldots > b_m > c$. Thus b_1 is the current Condorcet winner, and every candidate b_i beats c by $(m/3)-3$ votes.

Let V' contain an unregistered voter corresponding to each of the $S_j \in S$, with preference order $b_j^1 > b_j^2 > b_j^3 > c > \ldots$, where the first three entries correspond to the $b_i \in S_j$, and where entries after c are in arbitrary order. We

claim that there is a solution to X3C if and only if $m/3$ or fewer voters from V' can be added to V so that c becomes a Condorcet winner.

First assume that there exists an exact 3-cover. Including the corresponding voters in the election has the following effects: it increases the score of each b_i against c by exactly 1 vote, for a total score of $(m/3) - 2$; and it increases the score of c against each b_i by exactly $(m/3) - 1$ votes. Thus c becomes the Condorcet winner.

Now assume that c can be made the Condorcet winner by adding $m/3$ or fewer voters. There cannot be more than 1 added voter who prefers b_i to c, since then b_i would gain 2 or more votes for a total score against c of at least $(m/3)-1$; and c would gain no more than $(m/3)-2$ votes, and so would lose to b_i. Thus each b_i is preferred to c by either 0 or 1 added voters. If b_i is preferred by 1 added voter, then for c to win he must be preferred by $(m/3) - 1$ added voters; and since some voter must be added, there must be exactly $m/3$ added voters.

If there are no added voters who prefer b_i to c, then since the preferences of the $m/3$ added voters each include 3 positions above c, by the pigeon-hole principle there must be some other $b_{i'}$ that is ranked above c by more than 1 voter. This contradicts the requirement that no more than 1 added voter prefer any other candidate to c. Therefore each b_i is preferred to c by exactly 1 of the $m/3$ added voters. Thus the added voters correspond to an exact 3-cover of the b_i. □

Theorem 11. *Plurality voting is computationally vulnerable to control by deleting voters.*

Proof. Evaluate the current election and let the candidates who score no lower than c be $\{c_i\}$, where the score of c_i exceeds that of c by $n_i \geq 0$ votes. Now if $\sum(n_i + 1) > K$, then c cannot be made the unique plurality winner. If, on the other hand, $\sum(n_i + 1) \leq K$, then–if c received any votes at all–deleting for each i $n_i + 1$ voters who voted for c_i leaves c the unique plurality winner. This can be determined in $O(|V|)$ time. □

Theorem 12. *Condorcet voting is computationally resistant to control by deleting voters.*

Proof. We show that the problem is NP-hard by transformation from X3C.

For any instance of X3C create an election with candidates c and b_i ($i = 1, \ldots, m$). Let V consist of the following. To each S_j there corresponds a voter s_j who prefers candidates b_1^j, b_2^j, b_3^j to all others, and is otherwise indifferent. In addition, to each b_i there is a set of identical voters who prefer all other candidates to b_i, and are otherwise indifferent. For each b_i the number of such voters is equal to the number of S_j in which b_i appears. Thus in a pairwise election between c and any b_i–say b_k–b_k is preferred by only those voters s_j for which $b_k \in S_j$; but this vote is exactly offset by the equal number of b_k voters who prefer c to b_k. Thus c ties every other candidate, and so is not a Condorcet winner.

Suppose that c be made the Condorcet winner by deleting $m/3$ or fewer voters. To have become the Condorcet winner, c must have gained at least m votes in total to defeat the m candidates who previously tied him. Since deleting a b_i voter gains only 1 vote for c, while deleting an s_j voter gains 3 votes for c, all all of the deleted voters must be s_j voters. Furthermore, each candidate b_i must be favored by at least one of the deleted s_j voters, since otherwise the candidate b_i would continue to tie c. Moreover, no candidate can be favored by more than one of the deleted voters, since then some other candidate must fail to be favored by any of the deleted voters. Thus the S_j corresponding to the deleted voters must form an exact 3-cover of the b_i's.

Similarly, if there exists an exact 3-cover, then deleting the voters corresponding to the S_j of the cover reduces the vote for each b_i against c by 1, enabling c to defeat each b_i in pairwise election. Thus c can be made a Condorcet winner. □

Theorem 13. *Plurality voting is computationally vulnerable to control by partition of voters.*

Proof. We show how, in $O(|C||V|)$ time, to either construct a partition of V

into (V_1, V_2) that will result in victory for c, or else to conclude that no such partition exists.

To avoid the ambiguity of ties, we require that each subcommittee elect a single candidate to proceed to the larger election. Let n_i be the number of voters whose first choice is candidate i. Then the following two conditions are necessary and sufficient for control by partitioning candidates:

1. There exists at least one candidate beatable by c in a pairwise election (that is, there exists some candidate i for which $n_i < n_c$);

2. Let c' be the strongest candidate of those beatable by c (that is, c' is a candidate for which $n_{c'}$ is maximum among all candidates with $n_i < n_c$). Then, for every candidate i, $n_i \leq n_c + n_{c'} - 2$.

These conditions are sufficient since they enable us to construct the subcommittees as follows. Let V_1 initially contain all voters whose first choice is c, and let V_2 initially contain all voters whose first choice is c'. Then V_1 contains n_c votes for c, and V_2 contains $n_{c'}$ votes for c'. Now for each remaining candidate i, divide the n_i voters for i between V_1 and V_2 so that V_1 receives no more than $n_c - 1$ voters who prefer i, and V_2 receives no more than $n_{c'} - 1$ voters who prefer i. This preserves the fact that c will be elected by V_1 and c' will be elected by V_2 (after which c will defeat c' in the large election).

The first condition is necessary, since otherwise c would lose the large election, whoever his competitor. The second condition is necessary since, if there exists some candidate j with $n_j > n_c + n_{c'} - 2$, then–since we are disallowing ties–j must defeat c in either the subcommittee election or in the final election, depending on how the supporters of j are divided between the subcommittees.

The bottleneck in checking these conditions is the identification of c', which requires evaluating $O(|C|)$ elections, each of which requires $O(|V|)$ work. □

Theorem 14. *Condorcet voting is computationally resistant to control by partition of voters.*

Proof. We show that the problem is NP-hard by transformation from X3C, where without loss of generality we assume that each b_j appears in at least 3 of the S_i.

From an arbitrary instance of X3C, create an election with set of candidates $B \cup \{x, y, c\}$, and voters as follows.

1. There are 2 voters with preference order $c > B, x, y$; that is, they prefer c, and are indifferent among the remaining candidates.
2. There are $n + 1 - m/3$ voters with preference order $x > y > B, c$.
3. To each S_i there corresponds a voter with preference order $\{b_i^1, b_i^2, b_i^3\} = S_i > c, y, B - S_i > x$.

Now if there exists an exact 3-cover, construct a partition of V by letting V_1 contain the two voters who prefer c together with all the voters that correspond to an S_i in the cover. The remaining voters comprise V_2.

The Condorcet winner in V_1 is c: c defeats each candidate b_j since c gets 2 votes from the voters who prefer c to all others, while each b_j gets only 1 vote from all the remaining voters in V_1 (since each b_j is in exactly one S_j in the 3-cover). In addition, c defeats both x and y.

The Condorcet winner in V_2 is x: From voters with preferences $x > y > B, c$, x gets $n+1-m/3$ votes against each other candidate, while votes for any other candidate total no more than $n - m/3$ (since an exact 3-cover has cardinality m/3).

In the subsequent election among the full set of voters V, c is preferred to x by $2 + n$ voters while only $n + 1 - m/3$ voters prefer x, so that c wins the election. Thus, if there exists an exact 3-cover, then there exists a partition of the voters that result in victory for c.

Now we show that if there exists a partition that results in victory for c, then there must exist an exact 3-cover. Suppose that there exist V_1, V_2 and z such that $V_1 \cup V_2 = V$, $V_1 \cap V_2 = \emptyset$, $E(V_1, C) = \{c\}$, $E(V_2, C) = \{z\}$, and

$E(V, \{c, z\}) = \{c\}$. Then z must in fact be x, since each of the other candidates would defeat c in the final election.

We can assume that V_1 contains both of the voters who prefer c to all others, since altering V_1 and V_2 to enforce this can only help c in the election among the voters of V_1, and can only help x in the election among the voters of V_2. Similarly, we can assume that V_2 contains all the voters who prefer x.

Now consider how the voters corresponding to the S_i are distributed. No two S_i containing the same b_j can be in V_1 since then b_j would get 2 votes against c, and c would not be a (strict) Condorcet winner among the voters of V_1. However, in V_2 there are $n + 1 - m/3$ voters who prefer x, so V_2 cannot contain more than $n - m/3$ voters who prefer another candidate to x. Thus V_1 must contain $m/3$ disjoint S_i, which comprise an exact 3-cover. □

Finally, we note that all of the problems shown in this paper to be NP-hard, are also members of the problem class NP, and so are, more specifically, NP-complete.