Soc Choice Welfare (1989) 6:227–241

Social Choice and Welfare

The Computational Difficulty of Manipulating an Election*

J. J. Bartholdi III, C. A. Tovey, and M. A. Trick**

School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, GA 30332, USA

Received June 9, 1987 / Accepted July 29, 1988

Abstract. We show how computational complexity might protect the integrity of social choice. We exhibit a voting rule that efficiently computes winners but is computationally resistant to strategic manipulation. It is *NP*-complete for a manipulative voter to determine how to exploit knowledge of the preferences of others. In contrast, many standard voting schemes can be manipulated with only polynomial computational effort.

1. Introduction

We consider a voting scheme to be an algorithm that takes as imput a set C of candidates and a set V of preference orders that are strict (irreflexive and antisymmetric), transitive, and complete on C. The algorithm outputs a subset of C, who are the winners (allowing for ties).

Several celebrated theorems [4, 6, 14] show that any voting scheme that meets certain innocuous-looking rationality criteria must be susceptible to manipulation by strategic voting; that is, a voter with complete knowledge of the preferences of all the other voters can achieve a social choice more to his liking by misrepresenting his preferences.

Like the famous theorem of Arrow, these "impossibility" theorems fascinate because they seem to confound our ideals. They suggest that our methods of social choice must be either unfair or else inherently susceptible to abuse. The contribution

* Presented at Purdue University, March 1987; at the University of Arizona, April 1987; at Massachusetts Institute of Technology, April 1987; at Yale University, November 1987; at Centre International de Rencontres Mathematiques, Marseille-Luminy, April 1988. This research was supported in part by Presidential Young Investigator Awards from the National Science Foundation to the first two authors (ECS-8351313 and ECS-8451032), and by grant N00014-86-K-0173 from the Office of Naval Research.

** The authors appreciate many helpful comments and suggestions by the editor and three anonymous referees. We also thank Michel Balinski, Salvador Barbera, Jean-Pierre Barthelemy, and Peyton Young for stimulating discussions.

of this paper to suggest how such a gloomy prospect can be ameliorated on operational grounds: We exhibit a voting scheme that is computationally resistant to manipulation even though it is easy to compute the winner. While manipulable in principle, this scheme might not be manipulable in practice – even assuming free and perfect information – because of excessive computational requirements. This is different from the idea of [13] that degrees of manipulability be distinguished by their information requirements. Instead, we make distinctions based on the time required to process that information.

First we show that many of the well-known and historically used voting schemes can be "efficiently manipulated" in the sense that only polynomial time is required to determine how to vote so as to exploit knowledge of the preferences of other voters. Secondly, we prove that it is *NP*-complete to determine how to manipulate a scoring method that is in practical use (by, for example, the International Federation of Chess).

2. Computational Complexity

For those not familiar with computational comlexity, we provide a quick sketch of issues and terms. We urge the reader to consult [5] for more detail.

An algorithm is considered formally efficient if it requires a number of computational steps that is at most polynomial in the size of the problem. Problems for which there are polynomial-time algorithms are generally considered to be tractable, and those which require exponential time to solve are considered inherently intractable. Practical experience confirms the validity of this distinction.

One way to measure the difficulty of a problem is by the worst-case time to solve an instance of given size. Based on this measure, complexity theorists have identified a hierarchy of "complexity classes" into which problems might fall. For example, the problem class *NP* consists of those questions for which a "yes" answer can be justified in polynomial time, and the hardest problems in *NP* are known as "*NP*-complete". *NP*-complete problems are all equivalent in the sense that any problem in *NP* can be reworded as an instance of an *NP*-complete problem within polynomial time. Thus if there exists an algorithm to solve any *NP*-complete problem within guaranteed polynomial time, then all *NP*-complete problems could be solved by transforming them to instances of the easily solvable problem. Since no algorithm of guaranteed polynomial time has been found for any *NP*-complete problem, this is widely taken as evidence for their inherent intractability. To show that a problem is *NP*-complete it is enough to show that it is within the class *NP*, and that it is "at least as difficult as" some other problem known to be *NP*-complete. The first point is generally trivial to establish; the latter point can be established by showing how the known hard problem can be quickly converted to a special case of the problem in question. In showing that the two problems are "equivalently difficult", the argument is frequently rather formal, and with few concessions to intuition.

Classifying the complexity of voting problems is a natural refinement of previous work in social choice which has dealt with the distinction between the impossible and the possible [4, 6, 14], and, more recently, between the noncom-

putable and the computable [7, 9]. Here we distinguish between the intractable and the tractable, and conclude that, even if some electoral problems admit of a computable solution, that solution might be impractical. The solution might be impractical because the time required to compute a solution can increase exponentially in the size of the problem.

3. Many Voting Schemes are Easy to Manipulate

Consider a fixed voting scheme, and suppose that a manipulative voter knows in advance the preferences of every other voter; is there a preference ordering (possibly different from his true preferences) that the manipulator can adopt so that a specified candidate c is a winner? If a voter can answer this question, he can determine whether he can manipulate the election. We formalize this question as follows.

Existence of a Winning Preference

Instance: Set of candidates C, and a distinguished member c of C; set V of transitive preference orders on C.

Question: Does there exist a preference order P that will ensure that c will be the winner?

If this can be answered within polynomial time, then the voting scheme is "easily manipulable". We show that many commonly-used voting schemes are easily manipulable by the following simple procedure.

Algorithm Greedy-Manipulation

Input: preferences of all other voters, and a distinguished candidate c.

Output: either a preference order that, together with those of all the the other voters, will ensure that c is a winner, or else a claim that no such preference order exists.

Initialization: Place c at the top of the preference order.

Iterative Step. Determine whether any candidate can be placed in the next lower position (independent of other choices) without preventing c from winning. If so, place such a candidate in the next position; otherwise terminate claiming that c cannot win.

The following describes voting schemes that are manipulable by Greedy-Manipulation. Assume that the preferences of all the voters but the manipulator are fixed and known to the manipulator. For any preference P and candidates i, j, let iPj mean that i is preferred to j under P, so that $\{j : iPj\}$ is the set of candidates to whom i is preferred under P.

Theorem 1. *Greedy-Manipulation will find a preference order P that will make candidate c a winner (or conclude that it is impossible) for any voting scheme that can*

be represented as function $S(P): C \to \mathbf{R}$ *that is both*

– "*responsive*": *a candidate with the largest* $S(P, i)$ *is a winner; and*

– "*monotone*": *for any two preference orders* P *and* P' *and for any candidate* i, $\{j: iP'j\} \subseteq \{j: iPj\}$ *implies that* $S(P', i) \leqq S(P, i)$.

Proof. First note that if *Greedy-Manipulation* successfully constructs an order, that order will guarantee victory for candidate c. Now we show that if an order exists that ensures victory for c, then *Greedy-Manipulation* will construct it. Suppose *Greedy-Manipulation* terminates without having constructed a preference order, and let U be the set of unassigned candidates when *Greedy-Manipulation* terminates. Let P' be an order which would enable c to win, and let u be the highest ranked member of U under P'. Consider any completion P of the preference order started by *Greedy-Manipulation* that places u in the highest unassigned place. By responsiveness, $S(P', c) \geqq S(P', u)$, and because $\{j: uPj\}$ is contained in $\{j: uP'j\}$, $S(P', u) \geqq S(P, u)$. Furthermore, by the initialization of the algorithm and by monotonicity, $S(P, c) \geqq S(P', c)$. These inequalities imply $S(P, c) \geqq S(P, u)$. But, by the iterative step of *Greedy-Manipulation*, $S(P, c) < S(P, u)$ since u cannot go in the assigned slot. This is a contradiction. □

Corollary. *Any voting system that satisfies the conditions of Theorem 1, and for which* S *is evaluatable in polynomial time, can be manipulated in polynomial time.*

Proof. Greedy-Manipulation executes within polynomial time since no more than n iterations are required, and each iteration requires no more than n evaluations of S (by monotonicity of S) with each evaluation of S requiring only polynomial time (by assumption). □

Many voting schemes in common use satisfy the conditions of Theorem 1 and so are efficiently manipulable by *Greedy-Manipulation*. These include:

Plurality (Each voter casts 1 vote for their most preferred candidate). Function: Let b_i be the plurality score of candidate i among all voters except the manipulator; then $S(P, i) = b_i + 1$ if $|\{j: iPj\}| = |C| - 1$, else $= b_i$.

Positional (Borda count) (Each voter casts $|C|$ votes for their most preferred candidate, $|C| - 1$ votes for their next-most-preferred, ..., and 1 vote for their least preferred candidate.) Function: Let b_i be the positional score of candidate i among all voters except the manipulator; then $S(P, i) = b_i + |\{j: iPj\}| + 1$.

Maximin (A winner is a candidate who maximizes the minimum number of voters who prefer him to another candidate in pairwise elections.) Function: Let V_{ij} be the voters who prefer i to j; then $S(P, i) = \min_j(|V_{ij}| + 1$ if iPj; $|V_{ij}|$ if $jPi)$.

Copeland's method (A winner is a candidate who maximizes the number of victories minus the number of defeats in pairwise elections.) Function: $S(P, i) =$ (number of candidates that i beats in pairwise contests) – (number of candidates to whom i loses in pairwise contests) based on the preferences of all the voters, including the manipulator.

In addition, any monotone increasing function of such functions (which includes lexicographic combinations) still satisfies the hypotheses of Theorem 1.

Consequently, baroque scoring methods such as 1/3 times the positional count plus 1/5 times the Copeland count plus 2/7 times the maximin count are still efficiently manipulable by *Greedy-Manipulation.*

4. A Voting Scheme that is Computationally Resistant to Manipulation

The Copeland voting scheme ranks the candidates according to the number of pairwise contests they win minus the number they lose [11, 12]. When all candidates are compared against each other pairwise (so that they participate in the same number of contests), this is equivalent to scoring simply by the number of contests won. Many organizations use this method of social choice to rank contestants, and extend it by adding a tie-breaking rule that we refer to the "second-order Copeland scheme": In case of a tie, the winner is the candidate whose defeated competitors have the largest sum of Copeland scores.

(The Federation Internationale Des Echecs and the United States Chess Federation implement tie-breaking rules that are either identical to, or are minor variants of, the second-order Copeland scheme [8, 10]. For example, for round-robin tournaments under USCF rules, the primary score of each player is the number of opponents he has defeated plus one-half the number of opponents he has tied. In case of ties with respect to primary score, a secondary score is computed to be the sum of the primary scores of all the opponents he has defeated plus one-half the primary scores of all the opponents with whom he has drawn.)

It requires only polynomial time to compute the winner of an election under either Copeland or second-order Copeland schemes. Furthermore, by the theorem of Gardenfors [4] both schemes are manipulable in principle. However, we make an important distinction here: While a first-order Copeland scheme can be manipulated efficiently (by Theorem 1), second-order Copeland is computationally resistant to manipulation, as we will show. Specifically, it is *NP*-complete to determine whether one can misrepresent one's preferences to exploit knowledge of the preferences of others. (Note that second-order Copeland violates the hypothesis of monotonicity required by Theorem 1 for manipulability by *Greedy-Manipulation.*) Thus second-order Copeland circumvents the Gardenfors impossibility theorem on operational grounds: it satisfies the hypotheses-it is neutral, anonymous, and Condorcet – but it is computationally difficult to manipulate.

We can easily modify Second Order Copeland to similarly circumvent the Gibbard-Satterthwaite theorem [6, 14]: simply impose an arbitrary order on the candidates to break ties in their Second Order Copeland score; then the voting scheme is single-valued (but no longer neutral). By the Gibbard-Satterthwaite theorem, this modified scheme is in principle subject to manipulation but by our results it is computationally resistant to manipulation. More formally, the Gibbard-Satterthwaite theorem states

Theorem. *No social choice function is simultaneously*
(1) single-valued;
(2) non-dictatorial;
(3) non-manipulable.

In contrast, we prove the following.

Theorem. *There exists a social choice function (Second Order Copeland) that is simultaneously*

(1) single-valued;

(2) non-dictatorial;

(3) easy to compute, but computationally difficult to manipulate. Moreover, this social choice function is anonymous, Pareto optimal, and Condorcet.

Intuitively, it is difficult to construct a manipulative preference under Second Order Copeland because it is difficult to know where to place candidates in the preference. For example, placing a favored candidate at the top can unintentionally improve the scores of rivals because of second order effects in the scoring. This forces the manipulator to consider all the exponentially-many possible preference orders.

To capture the way in which second-order Copeland is likely to be used in practice, we exhibit an instance in which a set of candidates is tied under the primary score (Copeland), and the difficulty is to manipulate the tie-breaking score (second-order Copeland). The outline of the argument is as follows. First we show that a logic problem that is known to be hard can be embedded in a problem of scoring tournaments, so the tournament problem is hard. Then we show that the tournament problem can be embedded in *Existence of a Manipulative Preference for Second-Order Copeland*, which must therefore be hard.

A word of caution to the reader: The argument embeds a logic problem in a graph problem, and the graph problem in the voting problem. Since we are showing that these problems are in a sense equivalent, we variously adopt the elementary terminology of logic, graph theory, and voting. Where one field has a concept we need, we occasionally switch terminology mid-argument, rather than burden the reader with new but equivalent definitions.

Now we begin by showing that a problem of tournament scoring is hard. The tournament problem is illustrated by the final round of a round-robin chess tournament. The tournament requires each contestant to play every other contestant, but there is one more round (set of pairwise contests) to be played. The question is whether there exists a set of outcomes for the final round that will guarantee tournament victory for a particular competitor. We formalize this as:

Tournament Outcome

Instance: A complete simple graph (with vertices corresponding to candidates), for which each edge (i,j) can be either directed (corresponding to candidate i having beaten candidate j) or undirected (corresponding to the contest between candidates i and j not having been decided); a distinguished vertex (candidate) c.

Question: Is there a way of assigning directions to the currently undirected eges so that c is the winner?

Theorem 2. *Tournament outcome under second-order Copeland is NP-complete.*

Proof. The problem is in *NP* because we can quickly prove a "yes" answer by showing a set of outcomes and computing the second-order Copeland scores. We show the problem is as hard as an *NP*-complete problem by embedding within it 3,4-*Satisfiability*, which is known to be *NP*-complete [16].

3,4-SAT

Instance. An expression consisting of clauses $C_1, \ldots, C_m$ over variables $X_1, \ldots, X_n$, with each clause containing exactly 3 different variables, and each variable appearing in exactly 4 clauses.

Question. Is there a satisfying truth assignment for the set of clauses?

Given an instance of 3,4-SAT, construct a instance of *Tournament Outcome* as follows. Create a set of candidates corresponding to clauses $C_1, \ldots, C_m$ and to literals $X_1, \tilde{X}_1, \ldots, X_n, \tilde{X}_n$ (where $\tilde{X}$ is the negation of X). In addition, create a distinguished candidate c, whose victory is in question. All pairwise contests have been decided except those between the pairs of literal candidates X_j and $\tilde{X}_j$ $(j=1 \ldots n)$. In particular, each "clause candidate" C_i $(i=1 \ldots m)$ defeated the 3 "literal candidates" corresponding to the literals in the clause, and lost to all other candidates.

Model the candidates and their contests as a graph in which there is a vertex for each candidate. If candidate i has defeated candidate j, there is an edge directed from i to j; if the contest between candidates i and j has not yet been decided, connect i and j by an undirected edge. Consequently, in G each edge $(X_j, \tilde{X}_j)$ is undirected, and all other edges are directed from winner to loser. (See Fig. 1.) Let R be the set of undirected edges corresponding to remaining pairwise contests.

For the candidate corresponding to vertex v, his tentative scores $TS(v)$ and $TS^2(v)$ are his Copeland score and second-order Copeland score, respectively, not counting possible points from contests that have not yet been decided. Let $S(v, R)$ and $S^2(v, R)$ denote the eventual Copeland and second-order Copeland scores of v, depending on the outcomes of the remaining contests R. Note that for all C_i, $S(C_i, R) = TS(C_i)$ is independent of R.

In Appendix 1, we show how to pad graph G with additional candidates and assign outcomes to their pairwise contests so that the following properties hold:

Property 1. For any specification of outcomes for R, the candidates c and all C_i will be tied for first place with respect to (first-order) Copeland score; that is, $S(c, R) = S(C_1, R) = \ldots = S(C_m, R) > S(v, R)$ for all other v.

Property 2. The second-order Copeland score of the distinguished candidate c is independent of R. (Accordingly, we abbreviate $S^2(c, R)$ as $S^2(c)$.)

Property 3. Each candidate C_i $(i=1 \ldots m)$ has $TS^2(C_i) = S^2(c) - 3$.

Property 4. Each C_i $(i=1 \ldots m)$ defeated the 3 candidates corresponding to the literals that clause C_i contains in the instance of 3,4-SAT, and lost to all other candidates.

Property 5. For all outcomes R, $S^2(c) > S^2(v, R)$ for every v other than a clause candidate.

Now we claim that if G has Properties 2–5, then there is a set of outcomes for the remaining contests R which will make c the unique winner if-and-only-if the instance of 3,4-SAT is satisfiable. To see this, imagine candidate C_i $(i=1 \ldots m)$ during the last round of contests. His own contests are over, so his Copeland score

has been determined, and (by Property 3) he is currently 3 points short of a share of first place with c and possibly other C_j. By Property 4, C_i has lost to all but 3 of the literals, so the outcomes of only 3 contests (those containing the literals defeated by C_i) could improve his second-order score. If all 3 contests go as C_i wishes (the candidates that C_i defeated win their contests), then, by Properties 2 and 3, C_i's second-order score will equal that of c, and by Property 5 C_i will own a share of first place; otherwise, c will beat C_i. Interpreting a literal losing to its complement as the literal being set to TRUE in the instance of 3,4-SAT, candidate C_i will lose to c if-and-only-if clause C_i is satisfied. Thus satisfiability of the 3,4-SAT expression corresponds precisely to all of the C_i's ($i=1...m$) being defeated by c.

Corollary. *Tournament Manipulation under Copeland scoring with second-order Copeland tie-breaking is NP-complete.*

Proof. This follows from Property 1 of the proof of Theorem 2. □

Remark. This means that in a chess tournament it can be difficult for a team to play strategically when the tournament will be decided on tie-breaks. Furthermore, note that in the proof of Theorem 2, the undecided contests were candidate-disjoint; that is, no candidate was involved in more than one undecided contest. This means that it can be hard to manipulate even the final round of a tournament. We will require this fact in the proof of the next theorem, but first we need a technical lemma.

Let G be a complete graph on n vertices, where each edge may be directed or undirected. We show that there exists a set of $n(n-1)$ voters whose preferences produce the graph G of outcomes of pairwise contests between n candidates (where undirected edges signify ties). This set of voters also decide every non-tied contest by exactly 2 votes.

It is known that any complete directed graph of outcomes is realizable by a small number of voters [15]); our result differs only in allowing G to contain undirected edges representing ties, and in stipulating that the margin of victory be at least 2. All that is needed for our purposes is that G can be realized with a number of voters that is a polynomial in n.

Our argument depends on a theorem of elementary graph theory that the edges of K_n, the complete undirected graph on n vertices, can be partitioned into $(n-1)/2$ Hamiltonian cycles if n is odd [see, for example, p. 13 of 3].

Lemma. *Any set of outcomes (including ties) of simple majority pairwise contests between n candidates is realizable with $n(n-1)$ voters in such a way that every non-tied contest is decided by 2 votes.*

Proof. First assume n is odd and let $\{H_i\}$ be a set of $(n-1)/2$ edge-disjoint Hamiltonian paths on K_n. For each H_i create 2 sets of voters, V_i and W_i, as follows. Arbitrarily fix the vertex sequence of H_i to be $c_1, c_2, \ldots, c_n$, and define voter V_{ij} to have preferences $(c_j > c_{j+1} > \ldots > c_{j-1})$ (where the subscripts are understood to "wrap around" so that the successor of n is 1). The preferences of each W_{ij} are ecactly the opposite of V_{ij}. Thus we have created $2n$ voters corresponding to each of $(n-1)/2$ Hamiltonian paths, for a total of $n(n-1)$ voters. Furthermore, by symmetry, for every voter that prefers c_j to c_k, there exists an opposite voter who prefers c_k to c_j, so that every contest is tied.

Now consider any complete graph G on these n vertices, with each edge either directed or undirected. We will adjust the preferences of the voters to produce G. For each pair of candiates c_j and c_k, examine the edge joining c_j and c_k in G: if it is undirected, do nothing. Otherwise, assume the edge is directed from c_j to c_k. Since c_j and c_k must be adjacent in some Hamiltonian path H_i, there must be a voter whose preference order is $(\ldots > c_k > c_j)$; that is, c_j follows c_k at the very bottom of the preference order. Interchange c_k and c_j in this voter's preferences, so that now c_j defeats c_k in a pairwise contest by 2 votes. Since each interchange is between adjacent positions in a preference order, there can be no disruption of the results of other contests.

If n is even, add a dummy vertex (candidate) to G, with edges directed arbitrarily. Since the number of candidates is now odd, by the argument immediately preceding, there exists a set of voters for which the graph is realizable. Delete the dummy candidate from the preferences of these voters. □

Theorem 3. *Existence of a winning preference for second-order Copeland is NP-complete.*

Proof. The problem is in *NP* since the outcome of the election can be computed in polynomial time.

We prove difficulty by showing that the problem can contain the tournament manipulation problem just shown to be *NP*-complete. First recall that any complete graph of outcomes (including undirected edges to represent ties) is realizable by a small number of voters (where "small" means polynomial in the number of candidates) so that every non-tied contest is decided by 2 votes [15]. Apply this to the graph of outcomes of pairwise contests from the construction of Theorem 2, where we interpret the undirected edges as indicating a tie between the corresponding candidates in a simple vote count. Let V be a set of voters that realizes this graph. Assume that the manipulator knows these preferences, and now wonders whether there exists a preference he can claim that will make c a winner.

Since all contests corresponding to directed edges have been decided by 2 votes, the manipulator cannot affect these outcomes; he can affect only the contests that are currently tied. As we have previously observed, those contests are candidate-disjoint, so the manipulator can vote to achieve any of the 2^R possible sets of outcomes by simply ranking the candidates of each pair in desired relative order. However, to determine whether any of these sets of outcomes will make the distinguished candidate c a winner, the manipulator must solve the difficult instance of *Tournament Outcome* contrived in the previous theorem. □

Corollary. *Existence of a winning preference for first-order Copeland with second-order Copeland tie breaks is NP-complete.*

Proof. This follows from Theorem 3 and the corollary to Theorem 2. □

We can now establish the main result of this section, that it is computationally difficult to manipulate Second Order Copeland. Intuitively, this follows from Theorem 3: since it is difficult to determine a preference that ensures the election of a particular candidate, it is difficult to manipulate. More formally:

Theorem 4. *Manipulation of second-order Copeland is NP-complete.*

Proof. In the construction of Theorem 3, let the social choice function break ties in the order of preference $C_1, C_2, \ldots, C_m, c, \ldots$. Let the three "literal" candidates defeated by C_1 (whose existence is guaranteed by Property 4) be denoted L_1, L_2, and L_3. Suppose now that the sincere preferences of the manipulator are $c, C_1, L_1, L_2, L_3, \ldots$. Then if the manipulator votes sincerely, Properties 2–5 and the tie-breaking rule result in C_1 the social choice. Since the manipulator prefers only c to C_1, there is only one way to manipulate: to vote insincerely to make c the winner. But to recognize when this is possible is *NP*-complete by Theorem 3. □

Corollary. *Manipulation of first-order Copeland with second-order Copeland tie breaks is NP-complete.*

Finally we observe that second-order Copeland can be manipulated with (at least formal) efficiency if we restrict its use to elections in which there are not "too many" candidates. In fact this is true of any voting scheme for which the winner can be efficiently computed. The manipulator could in principle compute the outcome of the election for each of the $|C|!$ possible preference orders. If each evaluation requires only polynomial time, and if $|C|$ is restricted so that $|C|! = O(p(|V|))$ for some polynomial p of fixed degree, then the total effort is, strictly speaking, polynomial.

5. Conclusions

Computational complexity is a new criterion by which to evaluate methods of social choice. Worst-case behavior in this regard might be a practical consideration for some decision methods (see, for example, [1, 2]), just as is worst-case behavior with respect to formalized notions of fairness.

Methods of social choice should be easy to use but hard to abuse; that is, they should identify winners within polynomial time, but they should also be provably difficult to exploit. We have shown that Second Order Copeland satisfies these properties. Moreover, Second Order Copeland is a reasonable voting scheme in that it satisfies many appealing rationality criteria, including unanimity (the Pareto principle), neutrality, anonymity, and Condorcet-winner.

These issues are similar to those in cryptography, where both encryption and authorized deciphering should be easy, but unauthorized deciphering should be difficult. As for a cryptographic scheme, we would prefer to know that a voting scheme is *dependably* hard to abuse, rather than merely hard in the worst-case, as we have proved. Unfortunately, this sort of result seems beyond the reach of current complexity theory. However, we can appeal to practical experience, which confirms that *NP*-hard problems are difficult to solve.

There are several concerns that might be raised regarding our interpretation of these results. We state and discuss them below.

Concern: Complexity is not an issue because manipulation is not an issue: complete information about preferences is never available.

Discussion: Since we are trying to protect the mechanism of social choice, it is best to make the most conservative assumptions.

Concern: It might be that very few among all possible instances of an election are actually manipulable.

Discussion: True. Unfortunately, both the Gibbard-Satterthwaite and the Gardenfors theorems are worst-case theorems; that is, they claim that there exist instances of manipulable elections, but do not say how many. In the same way, our theorem is a worst-case theorem; it claims that some instances among the manipulable elections are hard to manipulate, but does not say how many. Thus our theorem is weak exactly where the Gibbard-Satterthwaite and Gardenfors theorems are weak.

Concern: It might be that very few among all real instances of an election are actually manipulable.

Discussion: True. Perhaps real elections have sufficiently special structure that makes them dependably easy to manipulate. For example is might be that real elections are small enough that exponentially-increasing work is not a deterrent to manipulation. It also might be that people's preferences are specially structured so that it is not hard to recognize when manipulation is possible. These are empirical questions that must be tested by experiment.

Concern: It might be that there are effective heuristics to manipulate an election even though manipulation is *NP*-complete.

Discussion: True. The existence of effective heuristics would weaken any practical import of our idea. It would be very interesting to find such heuristics.

Concern: These results do not "ameliorate" the Gibbard-Satterthwaite or Gardenfors theorems because the real meaning of these theorems is something else: that since there is an incentive to manipulate, researchers must consider game-theoretic issues.

Discussion: Game-theoretic issues are certainly important, but our result weakens this justification for studying them.

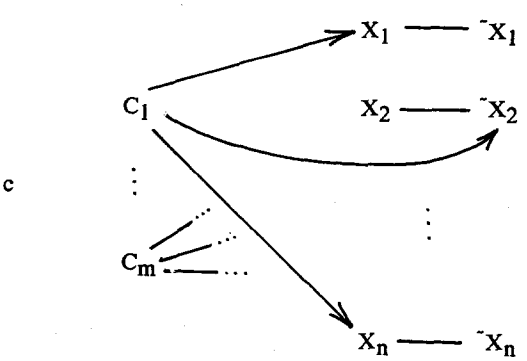

Fig. 1. Essential structure of the graph corresponding to an instance of 3,4-SAT. The undirected edges correspond to contests that have not yet been decided. Each C_i has an edge directed toward each of the 3 literals contained in clause C_i. (For clarity most of the directed edges are omitted, as are the candidates with which the graph is padded)

Appendix

We show how, given an instance of 3,4-SAT, to construct in polynomial time a graph satisfying Properties 1–5. Here is an overview of the construction of the

graph. G will contain three main types of vertices: those corresponding to clauses in the instance of 3,4-SAT; those corresponding to literals in the instance of 3,4-SAT; and "fillers", which pad the graph to make the scores suit our purpose. We will construct G so that all scores within each set are equal, or nearly so, but in pairwise contests clause candidates defeat fillers, fillers defeat literals, and literals defeat clauses. Finally, in addition to clause, literal, and filler vertices, there will be two distinguished vertices. One is C_0, corresponding to the contestant whose victory is in question, and the other is b, a "balancing" vertex that will enable us to adjust the second-order Copeland score of C_0.

Let m be the number of clauses and n the number of variables in the instance of 2,3-SAT. Note that $3m=4n$, so m is even and $m/2$ is an integer. Now define the vertices of G correspond to

- clauses: C_i $(i=1...m)$;
- literals: X_j, $\tilde{X}_j$ $(j=1...m)$;
- fillers: f_k $(k=1...30m)$;
- the contestant whose victory is in question: C_0;
- the balancing contestant: b.

Recall that G is a complete graph, so there is an edge between every pair of vertices. Each edge is directed according to the outcome of the pairwise contest between that pair of contestants, from the winner to the loser. Some contests will be as yet undecided, and so will be represented by undirected edges.

Orient the edges of G according to the following outcomes:

- contests between clauses and literals: Each literal defeats all clauses except the clauses containing it. Thus each literal defeats between $m-4$ and m clauses.
- contests between clauses and C_0: Each of the C_i $(i=0...m)$ wins $m/2$ and loses $m/2$ of these contests. This can be arranged by imagining the $m+1$ contestants C_i seated at every other chair around a symmetric $2(m+1)$-seat round table. Each C_i sits diametrically opposite an empty chair. Let each C_i defeat the $m/2$ people on his right, and lose to the $m/2$ people on his left.
- contests between literals: Each literal defeats exactly $n-1$ literals and loses to exactly $n-1$ other literals. This can be arranged by imagining the $2n$ literals seated at a round table with $2n$ seats, with each X_j seated diametrically opposite $\tilde{X}_j$. Let each literal defeat the first $n-1$ literals on his right (up to $\tilde{X}_j$), and lose to the first $n-1$ literals on his left (up to $\tilde{X}_j$). The contest between each X_j and $\tilde{X}_j$ has not yet been decided and so the corresponding edge is undirected.
- contests between fillers: Each f_k (k odd) defeats $15m-1$ other f_k $(k=1...30m)$. Each f_k (k even) defeats $15m$ other f_k $(k=1...30m)$. (It is this slight difference that allows us to achieve Property 3.) This can be arranged by imagining all the fillers seated around a $30m$-seat round table in the sequence $f_1, f_3, \ldots, f_{30m-1}$, $f_2, f_4, \ldots, f_{30m}$. Each f_k defeats the $15m-1$ fillers to his right, and loses to the $15m-1$ fillers to his left. Contests between fillers seated diametrically opposite each other are always between a filler of even index and a filler of odd index, and are won by the filler of even index.
- contests between literals and fillers: Every filler defeats every literal;
- contests between C_0 and literals: C_0 defeats X_1 and $\tilde{X}_1$, but loses to X_j and $\tilde{X}_j$ for $j=2...n$. (This gives Property 2.)
- contest between C_0 and b: C_0 defeats b;

– contests between b and clauses: b defeats all clauses;

– contests between C_0, clauses, and fillers: Define $f_{26i+1}, f_{26i+2}, \ldots, f_{26i+26}$ to be associates of C_i $(i=0 \ldots m)$. Thus each C_1 has 26 associates, no filler is an associate of more than one C_i, and nearly $4m$ fillers are not associates.

Now let $L(C_i)$ be the total number (counting repetitions) of clauses of which the literals in C_i are members, plus 1 if C_i contains X_1 or $\tilde{X}_1$. (For example, in the expression "$\{X_1$ or $\tilde{X}_2$ or $\tilde{X}_3\}$ and $\{\tilde{X}_2$ or $\tilde{X}_3$ or $X_4\}$", $L(C_1)=1+2+2=5$. We count C_0 as containing X_1 and $\tilde{X}_1$ so $L(C_0)=4+2=6$.) Since no literal appears in more than 4 clauses, and X_1 and $\tilde{X}_1$ do not appear in the same clause, we have that the number of literals in $C_i=3 \leqq L(C_i) \leqq (3)(4)+1=13$. $L(C_i)$ is the number of contests won by one of $C_0, C_1, \ldots, C_m$ against the three literals that comprise C_i. Thus C_i defeats all f_k which are either not an associate of C_i; or for which $k=26i+2, 26i+4, \ldots, 26i+2L(C_i)$; or for which $k=26i+1, 26i+3, \ldots, 26i+2(13-L(C_i))-1$. In other words, C_i defeats all of its non-associates, and 13 of its 26 associates ($L(C_i)$ of the 13 defeated associates being fillers of even index.)

– contests between b and f_k: b loses to all but the last $n+4$ fillers; that is, b loses to f_k for $k=1, 2, \ldots, 27m, \ldots, 30m-n-4$, but defeats f_k for $k=30m-n-3, \ldots, 30m$.

– contests between b and literals: b loses to all literals.

Verifying that Properties 1–5 are satisfied consists entirely of arithmetic computation. First, we compute the first-order Copeland scores of some of the candidates:

(1) $S(C_i)$ $(i=1 \ldots m)=m/2+3+0+(30m-13)=S(C_0)=m/2+2+1+(30m-13)$, where the terms are due to victories over contestants from $C_0 \ldots C_m$, literals, B, and fillers, respectively.

(2) $S(B)=m+0+0+(n+4)=m+N+4$, where the terms are due to victories over contestants from $C_1 \ldots C_m$, C_0, literals, and fillers, respectively.

We do not know the Copeland scores of the literals, but we can compute their tentative scores. For any literal X_i,

$TS(X_i)=\{1$ if $i>1$; 0 if $i=1\}+m-\{\#$ of clauses in which X_i appears$\}+(n-1)+0+1$, where the terms are due to victories over C_0, clauses, literals, fillers, and B, respectively.

$S(X_i, R)=TS(X_i)$ or $TS(X_i)+1$. Thus

(3) $TS(X_i)=m+n+1-\{$the contribution of X_i to $L(\,)\}$

(4) If F_i is an associate, $S(F_i)=(15m-1)+\{1$ if i is even, else $0\}+1+2n+\{1$ if it beats its associate$\}$, where the terms are due to victories over fillers, B, and literals, respectively.

Now we verify Properties 1–5.

Property 1. The total number of nodes in G is $1+m+2n+30m+1=65m/2+2$; B loses to at least $26m$ fillers; each filler loses to at least $14m$ fillers; each literal loses to at least $30m$ fillers; hence for all v not a C_i, $S(v) \leqq 32(m/2)+2-14m < S(C_0)$, which verifies Property 1.

Property 2. Since for all i, C_0 either beats both X_i and $\tilde{X}_i$ or loses to both these literals, it follows that $S^2(C_0, R)$ is independent of R. This is Property 2.

Property 3. We compare the second order Copeland scores of the C_i and verify that for any i, $1 \leqq i < m$, $TS^2(C_i) = TS^2(C_{i+1})$. Define $TS^2(C_i) - TS^2(C_{i+1}) = \text{Diff}_C + \text{Diff}_B + \text{Diff}_L + \text{Diff}_F$, where the "Diff" terms are the differences in the contributions from the different classes of nodes, C, B, Literal, Filler. (Note: C includes C_0 as well as all the clause nodes.)

$\text{Diff}_C = 0$ since C_i and C_{i+1} both beat $m/2$ in class C and by (1) all class C members have identical value of $S(\,)$.

$\text{Diff}_B = 0$ since they both beat B.

$\text{Diff}_L = 3(m+n+1) - L(C_i) - [3(m+n+1) - L(C_{i+1})] = L(C_{i+1}) - L(C_i)$

$\text{Diff}_F = 0$ from non-associate fillers, which they both beat

$+|\{\text{evens } C_i \text{ beats}\}| - |\{\text{evens } C_{i+1} \text{ beats}\}|$ by (4)

$+13$ for C_i beating of C_{i+1}'s associates that beat it

-13 for C_{i+1} beating 13 of C_i's associates that beat it by (4)

$= L(C_i) - L(C_{i+1})$

Thus all the Diff terms cancel out and $TS^2(C_i)$ equals $TS^2(C_{i+1})$. This proves half of Property 3, that all the clause nodes have the same TS^2 score.

Now compare $TS^2(C_1)$ with $S^2(C_0)$. As before let $S^2(C_0) - TS^2(C_1) = \text{Diff}_C + \text{Diff}_B + \text{Diff}_L + \text{Diff}_F$.

$\text{Diff}_C = 0$ as before;

$\text{Diff}_B = m + (n+4)$ by (2);

$\text{Diff}_L = 2(n+m+1) - (\text{contribution of } X_1 \text{ to } L(\,) + \text{contribution of } \tilde{X}_1 \text{ to } L(\,) - 3(n+m+1) + L(C_1)$ by (3).

$= (n+m+1) + L(C_1) - L(C_0)$;

$\text{Diff}_F = L(C_0) - L(C_1)$ as before.

This Diff_F is cancelled out by part of Diff_L, and the sum of the differences equals $m + (n+4) - (n+m+1) = 3$. (Now it is clear why b beats $(n+4)$ fillers.) This verifies Property 3.

Property 4: The construction of the edges between C_i and the literals gives Property 4.

Property 5: To verify Property 5 we need only a rough estimate of the second order Copeland scores. By (4), for any C_i, the portion of the second-order score coming from the fillers is about $30m(15m+2n) > 480m^2$. Since the total number of nodes is only $65m/2 + 2$, this is not a bad estimate. Moreover, as shown in the verification of Property 1, every other node loses to at least $14m$ fillers. Hence the filler portion of other second order scores will be smaller by at least $14m(15m+2n) > 225m^2$. This is too large a deficit to overcome by contributions from the remaining $2(m/2)$ nodes, which cannot possibly contribute more than $2(m/2)(32m/2) > 81m^2$. Hence, every other node has smaller second order Copeland score than the C_i (smaller by more than $100m^2$). This verifies Property 5.

Thus the election we have constructed satisfies the properties claimed.

References

1. Bartholdi JJ III, Tovey CA, Trick MA (1989) Voting schemes for which it can be difficult to tell who won the election. Soc Choice Welfare 6:157–165
2. Bartholdi JJ III, Tovey CA, Trick MA (1987) How hard is it to control an election? Econometrica (submitted)
3. Bollobas, B (1979) Graph theory. Graduate Texts 63. Springer, Berlin Heidelberg New York
4. Gardenfors P (1976) Manipulation of social choice functions. J Econ Theory 13:217–228
5. Garey M, Johnson D (1979) Computers and intractability: a guide to the theory of *NP*-completeness. WH Freeman, San Francisco
6. Gibbard A (1973) Manipulation of voting schemes. Econometrica 41:587–601
7. Gottinger HW (1987) Choice and complexity. Math Soc Sci 14:1–17
8. Kazic B, Keene RD, Lim KA (eds) (1986) The official laws of chess. Maxmillan, New York
9. Lewis A (1985) On the effectively computable realizations of choice functions. Math Soc Sci 10: 43–80
10. Morrison M (ed) (1978) Official rules of chess, 2nd Edn. David McKay New York
11. Niemi RG, Riker WH (1976) The choice of voting systems. Sci Am 234:21–27
12. Nurmi H (1983) Voting procedures: a summary analysis. Br J Polit Sci 13:181–208
13. Nurmi H (1986) Problems of finding optimal voting and representation systems. E J Oper Res 24:91–98
14. Satterthwaite MA (1975) Strategy-proofness and Arrow's conditions. J Econ Theory 10:187–217
15. Stearns R (1959) The voting problem. Am Math Mon 66:761–763
16. Tovey CA (1984) A simplified *NP*-complete satisfiability problem. Disc Appl Math 8:85–89