Junta Distributions and the Average-Case Complexity of Manipulating Elections

Ariel D. Procaccia Jeffrey S. Rosenschein
School of Engineering and Computer Science
The Hebrew University of Jerusalem
Jerusalem, Israel
{arielpro, jeff}@cs.huji.ac.il

ABSTRACT

Encouraging voters to truthfully reveal their preferences in an election has long been an important issue. Previous studies have shown that some voting protocols are hard to manipulate, but predictably used $\mathcal{NP}$-hardness as the complexity measure. Such a *worst-case* analysis may be an insufficient guarantee of resistance to manipulation.

Indeed, we demonstrate that $\mathcal{NP}$-hard manipulations may be tractable in the *average-case*. For this purpose, we augment the existing theory of average-case complexity with new concepts; we consider elections distributed with respect to *junta distributions*, which concentrate on hard instances, and introduce a notion of *heuristic* polynomial time. We use our techniques to prove that a family of important voting protocols is susceptible to manipulation by coalitions, when the number of candidates is constant.

Categories and Subject Descriptors

F.2 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity;
I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent Systems*;
J.4 [**Computer Applications**]: Social and Behavioral Sciences—*Economics*

General Terms

Algorithms, Theory, Economics

Keywords

Computational complexity, Voting

1. INTRODUCTION

In multiagent environments, it may be the case that different agents have diverse preferences, and it is therefore important to find a way to aggregate agent preferences. A general scheme for preference aggregation is *voting*: the agents reveal their preferences by ranking a set of candidates, and a winner is determined according to a voting protocol. The candidates can be various entities such as beliefs or plans, and indeed may be potential real-life parliament members.

Things are made complicated by the fact that in many settings (as in reality) the agents are self-interested. Such an agent may reveal its preferences untruthfully, if it believes this would make the final outcome of the elections more favorable for it. Consequently, the outcome may be one that does not maximize social welfare. This problem is provably acute: it is known [8, 10] that, for elections with three or more candidates, in any voting protocol that is non-dictatorial,[1] there are elections where an agent is better off by voting untruthfully.

Fortunately, it is reasonable to make the assumption that the agents are computationally bounded. Therefore, although in principle an agent may be able to manipulate an election, the computation required may be infeasible. This has motivated researchers to study the computational complexity of manipulating voting protocols. It has long been known [3] that there are voting protocols that are $\mathcal{NP}$-hard to manipulate by a single voter. Recent results by Conitzer and Sandholm [5, 4] show that some manipulations of common voting protocols are $\mathcal{NP}$-hard, even for a small number of candidates. Moreover, in [6], it is shown that adding a pre-round to some voting protocols can make manipulations hard (even $\mathcal{PSPACE}$-hard in some cases). Elkind and Lipmaa [7] show that the notion of pre-round, together with one-way functions, can be used to construct protocols that are hard to manipulate even by a large minority fraction of the voters.

In Computer Science, the notion of hardness is usually considered in the sense of worst-case complexity. Not surprisingly, most results on the complexity of manipulation use $\mathcal{NP}$-hardness as the complexity measure. However, it may still be the case that most instances of the problem are easy to manipulate.

A relatively little-known theory of average case complexity exists [11]; that theory introduces the concept of distributional problems, and defines what a reduction between distributional problems is. It is also known that average-case complete problems exist (albeit artificial ones, such as a distributional version of the halting problem).

Sadly, it is very difficult to show that a certain problem is average-case complete, and such results are known only for a handful of problems. Additionally, the goal of the

[1] In a dictatorial protocol, there is an agent that dictates the outcome regardless of the others' choices.

AAMAS'06 May 8–12 2006, Hakodate, Hokkaido, Japan.

existing theory is to define when a problem is *hard* in the average-case; it does not provide criteria for deciding when a problem is *easy*. A step towards showing that a manipulation is easy on average was made in [7]. It involves an analysis of the plurality protocol with a pre-round, but focuses on a very specific distribution, which does not satisfy some basic desiderata as to what properties an "interesting" distribution should have.

In this paper, we engage in a novel average-case analysis, based on criteria we propose. Coming up with an "interesting" distribution of problem instances with respect to which the average-case complexity is computed is a difficult task, and the solution may be controversial. We analyze problems whose instances are distributed with respect to a *junta distribution*. Such a distribution must satisfy several conditions, which (arguably) guarantee that it focuses on instances that are harder to manipulate. We consider a protocol to be *susceptible* to manipulation when there is a polynomial time algorithm that can usually manipulate it: the probability of failure (when the instances are distributed according to a junta distribution) must be inverse-polynomial. Such an algorithm is known as a *heuristic* polynomial time algorithm.

We use these new methods to prove our main result: an important family of protocols, called *scoring* protocols, is susceptible to coalitional manipulation when the number of candidates is constant. Specifically, we contemplate *sensitive* scoring protocols, which include such well-known protocols as Borda and Veto. To accomplish this task, we define a natural distribution μ^* over the instances of a well-defined coalitional manipulation problem, and show that this is a junta distribution. Furthermore, we present the manipulation algorithm GREEDY, and prove that it usually succeeds with respect to μ^*.

We also show that all protocols are susceptible to a certain setting of manipulation, where the manipulator is unsure about the others' votes. This result depends upon a basic conjecture regarding junta distributions, but also has implications that transcend our specific definition of these distributions.

In Section 2, we outline some important voting protocols, and properly define the manipulation problems we shall discuss. In Section 3, we formally introduce the tools for our average case analysis: junta distributions, heuristic polynomial time, and susceptibility to manipulations. In Section 4 we prove our main result: sensitive scoring protocols are susceptible to coalitional manipulation with few candidates. In Section 5, we discuss the case when a single manipulator is unsure about the other voters' votes. Finally, in Section 6, we present conclusions and directions for future research.

2. PRELIMINARIES

We first describe some common voting protocols and formally define the manipulation problems with which we shall deal. Next, we introduce a useful lemma from probability theory.

2.1 Elections and Manipulations

An election consists of a set C of m candidates, and a set V of n voters, who provide a total order on the candidates. An election also includes a winner determination function from the set of all possible combinations of votes to C. We note that throughout this paper, $m = O(1)$, so the complexity results are in terms of n.

Different voting protocols are distinguished by their winner determination functions. The protocols we shall discuss are:

- *Scoring protocols:* A scoring protocol is defined by vector $\vec{\alpha} = \langle \alpha_1, \alpha_2, \ldots, \alpha_m \rangle$, such that $\alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_m$ and $\alpha_i \in \mathbb{N} \cup \{0\}$. A candidate receives α_i points for each voter which ranks it in the i'th place. Examples of scoring protocols are:
  - *Plurality:* $\vec{\alpha} = \langle 1, 0, \ldots, 0, 0 \rangle$.
  - *Veto:* $\vec{\alpha} = \langle 1, 1, \ldots, 1, 0 \rangle$.
  - *Borda:* $\vec{\alpha} = \langle m-1, m-2, \ldots, 1, 0 \rangle$.
- *Copeland:* For each possible pair of candidates, simulate an election; a candidate wins such a pairwise election if more voters prefer it over the opponent. A candidate gets 1 point for each pairwise election it wins, and -1 for each pairwise election it loses.
- *Maximin:* A candidate's score in a pairwise election is the number of voters that prefer it over the opponent. The winner is the candidate whose minimum score over all pairwise elections is highest.
- *Single Transferable Vote (STV):* The election proceeds in rounds. In each round, the candidate's score is the number of voters that rank it highest among the remaining candidates; the candidate with the lowest score is eliminated.

REMARK 1. We assume that tie-breaking is always adversarial to the manipulator.[2]

In the case of weighted votes, a voter with weight $k \in \mathbb{N}$ is naturally regarded as k voters who vote unanimously. In this paper, we consider weights in $[0, 1]$. This is equivalent, since any set of integer weights in the range $1, \ldots, \text{poly}n$ can be scaled down to weights in the segment $[0, 1]$ with $O(\log n)$ bits of precision.

The main results of the paper focus on scoring protocols. We shall require the following definition:

DEFINITION 1. Let P be a scoring protocol with parameters $\vec{\alpha} = \langle \alpha_1, \alpha_2, \ldots, \alpha_m \rangle$. We say that P is *sensitive* iff $\alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_{m-1} > \alpha_m = 0$ (notice the strict inequality on the right).

In particular, Borda and Veto are sensitive scoring protocols.

REMARK 2. Generally, from any scoring protocol with $\alpha_{m-1} > \alpha_m$, an equivalent sensitive scoring protocol can be obtained by subtracting α_m on a coordinate-by-coordinate basis from the vector $\vec{\alpha}$. Moreover, observe that if a protocol is a scoring protocol but is not sensitive, and $\alpha_m = 0$, then $\alpha_{m-1} = 0$. In this case, for three candidates it is equivalent to the plurality protocol, for which most manipulations are tractable even in the worst-case. Therefore, it is sufficient to restrict our results to sensitive scoring protocols.

[2] This is a standard assumption, also made, for example, in [5, 4].

We next consider some types of manipulations, state the appropriate complexity results, and introduce some notations.

Remark 3. We discuss the *constructive* cases, where the goal is trying to make a candidate win, as opposed to *destructive* manipulation, where the goal is to make a candidate lose. Constructive manipulations are always at least as hard (in the worst-case sense) as their destructive counterparts, and in some cases strictly harder (if one is able to determine whether p can be made to win, one can also ask whether any of the other $m-1$ candidates can be made to win, thus making p lose).

Definition 2. In the Individual-Manipulation problem, we are given all the other votes, and a preferred candidate p. We are asked whether there is a way for the manipulator to cast its vote so that p wins.

Bartholdi and Orlin [3] show that IM is $\mathcal{NP}$-complete in Single Transferable Vote, provided the number of candidates is unbounded. However, the problem is in $\mathcal{P}$ for most voting schemes, and hence will not be studied here.

Definition 3. In the Coalitional-Weighted-Manipulation (CWM) problem, we are given a set of weighted votes S, the weights of a set of votes T which have not been cast, and a preferred candidate p. We are asked whether there is a way to cast the votes in T so that p wins the election.

We know [5, 4] that CWM is $\mathcal{NP}$-complete in Borda, Veto and Single Transferable Vote, even with 3 candidates, and in Maximin and Copeland with at least 4 candidates.

The CWM version that we shall analyze, which is specifically tailored for scoring protocols, is a slightly modified version whose analysis is more straightforward:

Definition 4. In the Scoring-Coalitional-Weighted-Manipulation (SCWM) problem, we are given an initial score $S[c]$ for each candidate c, the weights of a set of votes T which have not been cast, and a preferred candidate p. We are asked whether there is a way to cast the votes in T so that p wins the election.

$S[c]$ can be interpreted as c's total score from the votes in S. However, we do not require that there exist a combination of votes that actually induces $S[c]$ for all c.

Definition 5. In the Uncertain-Votes-Weighted-Evaluation (UVWE) problem, we are given a weight for each voter, a distribution over all the votes, a candidate p, and a number $r \in [0, 1]$. We are asked whether the probability of p winning is greater than r.

Definition 6. In the Uncertain-Votes-Weighted-Manipulation (UVWM) problem, we are given a single manipulative voter with a weight, weights for all other voters, a distribution over all the others' votes, a candidate p, and a number r, where $r \in [0, 1]$. We are asked whether the manipulator can cast its vote so that p wins with probability greater than r.

If CWM is $\mathcal{NP}$-hard in a protocol, then UVWE and UVWM are also $\mathcal{NP}$-hard in it [5]. These problems will be studied in Section 5. We make the assumption that the given distributions over the others' votes can be sampled in polynomial time.

2.2 Chernoff's Bounds

The following lemma will be of much use later on. Informally, it states that the average of independent identically distributed (i.i.d.) random variables is almost always close to the expectation.

Lemma 1 (Chernoff's Bounds). *Let $X_1, \ldots, X_t$ be i.i.d. random variables such that $a \le X_i \le b$ and $\mathrm{E}[X_i] = \mu$. Then for any $\epsilon > 0$, it holds that:*

- $\Pr[\frac{1}{t}\sum_{i=1}^{t} X_i \ge \mu + \epsilon] \le e^{-2t\frac{\epsilon^2}{(b-a)^2}}$
- $\Pr[\frac{1}{t}\sum_{i=1}^{t} X_i \le \mu - \epsilon] \le e^{-2t\frac{\epsilon^2}{(b-a)^2}}$

3. JUNTA DISTRIBUTIONS AND SUSCEPTIBLE MECHANISMS

In this section we lay the mathematical foundations required for an average-case analysis of the complexity of manipulations. All of the definitions are as general as possible; they can be applied to the manipulation of any mechanism, not merely to the manipulation of voting protocols.

We describe a distribution over the instances of a problem as a collection of distributions $\mu_1, \ldots, \mu_n, \ldots$, where μ_n is a distribution over the instances x such that $|x| = n$. We wish to analyze problems whose instances are distributed with respect to a distribution which focuses on hard-to-manipulate instances. Ideally, we would like to insure that if one manages to produce an algorithm which can usually manipulate instances according to this distinguished "difficult" distribution, the algorithm would also usually succeed when the instances are distributed with respect to most other reasonable distributions.

Definition 7. Let $\mu = \{\mu_n\}_{n\in\mathbb{N}}$ be a distribution over the possible instances of an $\mathcal{NP}$-hard manipulation problem M. μ is a *junta* distribution if and only if μ has the following properties:

1. Hardness: The restriction of M to μ is the manipulation problem whose possible instances are only:
$$\bigcup_{n\in\mathbb{N}} \{x : |x| = n \wedge \mu_n(x) > 0\}.$$
Deciding this restricted problem is still $\mathcal{NP}$-hard.
2. Balance: There exist a constant $c > 1$ and $N \in \mathbb{N}$ such that for all $n \ge N$:
$$\frac{1}{c} \le \ \Pr_{x\sim\mu_n}[M(x) = 1] \le \ 1 - \frac{1}{c}.$$
3. Dichotomy: for all n and instances x such that $|x| = n$:
$$\mu_n(x) \ge 2^{-\mathrm{poly}n} \vee \mu_n(x) = 0.$$

If M is a voting manipulation problem, we also require the following property:

4. Symmetry: Let v be a voter whose vote is given, let $c_1, c_2 \ne p$ be two candidates, and let $i \in \{1, \ldots, m\}$. The probability that v ranks c_1 in the i'th place is the same as the probability that v ranks c_2 in the i'th place.

If M is a coalitional manipulation problem, we also require the following property:

5. Refinement: Let x be an instance such that $|x| = n$ and $\mu_n(x) > 0$; if all colluders voted identically, then p would not be elected.

The name "junta distribution" comes from the idea that in such a distribution, relatively few "powerful" and difficult instances represent all the other problem instances. Alternatively, our intent is to have a few problematic distributions (the family of junta distributions) convincingly represent all other distributions with respect to the average-case analysis.

The first three properties are basic, and are relevant to problems of manipulating any mechanism. The definition is modular, and additional properties may be added on top of the basic three, in case one wishes to analyze a mechanism which is not a voting protocol.

The exact choice of properties is of extreme importance (and, as we mentioned above, may be arguable). We shall briefly explain our choices. Hardness is meant to insure that the junta distribution contains hard instances. Balance guarantees that a trivial algorithm which always accepts (or always rejects) has a significant chance of failure. The dichotomy property helps in preventing situations where the distribution gives a (positive but) negligible probability to all the hard instances, and a high probability to several easy instances.

We now examine the properties that are specific to manipulation problems. The necessity of symmetry is best explained by an example. Consider CWM in STV with $m \geq 3$. One could design a distribution where p wins if and only if a distinguished candidate loses the first round. Such a distribution could be tailored to satisfy the other conditions, but misses many of the hard instances. In the context of SCWM, we interpret symmetry in the following way: for every two candidates $c_1, c_2 \neq p$ and $y \in \mathbb{R}$,

$$\Pr_{x\sim\mu_n}[S[c_1] = y] = \Pr_{x\sim\mu_n}[S[c_2] = y].$$

Refinement is less important than the other four properties, but seems to help in concentrating the probability on hard instances. Observe that refinement is only relevant to coalitional manipulation; we believe that in the analysis of individual voting manipulation problems, the first four properties are sufficient.

Definition 8. [11] A *distributional problem* is a pair $\langle L, \mu\rangle$ where L is a decision problem and μ is a distribution over the set $\{0,1\}^*$ of possible inputs.

Informally, an algorithm is a heuristic polynomial time algorithm for a distributional problem if it runs in polynomial time, and fails only on a small fraction of the inputs. We now give a formal definition; this definition is inspired by [11] (there the same name is used for a somewhat different definition).

Definition 9. Let M be a manipulation problem and let $\langle M, \mu\rangle$ be a distributional problem.

1. An algorithm A is a *deterministic heuristic polynomial time* algorithm for the distributional manipulation problem $\langle M, \mu\rangle$ if A always runs in polynomial time, and there exists a polynomial p and $N \in \mathbb{N}$ such that for all $n \geq N$:
$$\Pr_{x\sim\mu^n}[A(x) \neq M(x)] < \frac{1}{p(n)}. \qquad (1)$$

2. Let A be a probabilistic algorithm, which uses a random string s. A is a *probabilistic heuristic polynomial time* algorithm for the distributional manipulation problem $\langle M, \mu\rangle$ if A always runs in polynomial time, and there exists a polynomial p and $N \in \mathbb{N}$ such that for all $n \geq N$:
$$\Pr_{x\sim\mu^n,s}[A(x) \neq M(x)] < \frac{1}{p(n)}. \qquad (2)$$

Probabilistic algorithms have two potential sources of failure: an unfortunate choice of input, or an unfortunate choice of random string s. The success or failure of deterministic algorithms depends only on the choice of input.

We now combine all the definitions introduced in this section in an attempt to establish when a mechanism is susceptible to manipulation in the average case. The following definition abuses notation a bit: M is both used to refer to the manipulation itself, and the corresponding decision problem.

Definition 10. We say that a mechanism is *susceptible* to a manipulation M if there exists a junta distribution μ, such that there exists a deterministic/probabilistic heuristic polynomial time algorithm for $\langle M, \mu\rangle$.

4. SUSCEPTIBILITY TO SCWM

Recall [5, 4] that in Borda and Veto, CWM is $\mathcal{NP}$-hard, even with 3 candidates. Since Borda and Veto are examples of sensitive scoring protocols, we would like to know how resistant this family of protocols really is with respect to coalitional manipulation. In this section we use the methods from the previous section to prove our main result:

Theorem 1. *Let P be a sensitive scoring protocol. Then P, with candidates $C = \{p, c_1, \ldots, c_m\}$, $m = O(1)$, is susceptible to SCWM.*

Intuitively, the instances of CWM (or SCWM) which are hard are those that require a very specific partitioning of the voters in T to subsets, where each subset votes unanimously. These instances are rare in any reasonable distribution; this insight will ultimately yield the theorem.

The following proposition generalizes Theorem 1 of [5] and Theorem 2 of [4], and justifies our focus on the family of sensitive scoring protocols. A stronger version of Proposition 2 has been independently proven in [9]. Nevertheless, we include our proof, since it will be required in proving the hardness property of a junta distribution we shall design.

Proposition 2. *Let P be a sensitive scoring protocol. Then CWM in P is $\mathcal{NP}$-hard, even with 3 candidates.*

Definition 11. In the Partition problem, we are given a set of integers $\{k_i\}_{i\in[t]}$, summing to $2K$, and are asked whether a subset of these integers sum to K.

It is well-known that Partition is $\mathcal{NP}$-complete.

Proof of Proposition 2. We reduce an arbitrary instance of Partition to the following CWM instance. There are 3 candidates, a, b, and p. In S, there are $K(4\alpha_1 - 2\alpha_2) - 1$ voters voting $a \succ b \succ p$, and $K(4\alpha_1 - 2\alpha_2) - 1$ voters voting $b \succ a \succ p$. In T, for every k_i there is a vote of weight $2(\alpha_1 + \alpha_2)k_i$. Observe that from S, both a and b get $(K(4\alpha_1 - 2\alpha_2) - 1)(\alpha_1 + \alpha_2)$ points.

Assume first that a partition exists. Let the voters in T in one half of the partition vote $p \succ a \succ b$, and let the other half vote $p \succ b \succ a$. By this vote, a and b each have

$$\begin{aligned}&(K(4\alpha_1 - 2\alpha_2) - 1)(\alpha_1 + \alpha_2) + 2K(\alpha_1 + \alpha_2)\alpha_2\\ &= (\alpha_1 + \alpha_2)(4K\alpha_1 - 1)\end{aligned}$$

votes, while p has $(\alpha_1 + \alpha_2)4K\alpha_1$ points; thus there is a manipulation.

Conversely, assume that a manipulation exists. Clearly there must exist a manipulation where all the voters in T vote either $p \succ a \succ b$ or $p \succ b \succ a$, because the colluders do not gain anything by not placing p at the top in a scoring protocol. In this manipulation, p has $(\alpha_1 + \alpha_2)4K\alpha_1$ points, while a and b already have $(K(4\alpha_1 - 2\alpha_2) - 1)(\alpha_1 + \alpha_2)$ points from S. Therefore, a and b must gain less than $(2\alpha_2 K + 1)(\alpha_1 + \alpha_2)$ points from the voters in T. Each voter corresponding to k_i contributes $2(\alpha_1 + \alpha_2)\alpha_2 k_i$ points; it follows that the sum of the k_i corresponding to the voters voting $p \succ a \succ b$ is less than $K + \frac{1}{2\alpha_2}$, and likewise for the voters voting $p \succ b \succ a$. Equivalently, the sum can be at most K, since all k_i are integers and $\alpha_2 \geq 1$. In both cases the sum must be at most K; hence, this is a partition. □

Since an instance of CWM can be translated to an instance of SCWM in the obvious way, we have:

Corollary 3. *Let P be a sensitive scoring protocol. It holds that SCWM in P is $\mathcal{NP}$-hard, even with 3 candidates.*

4.1 A Junta Distribution

Let $w(v)$ denote the weight of voter v, and let W denote the total weight of the votes in T; P is a sensitive scoring protocol. We denote $|T| = n$: the size of T is the size of the instance.

Consider a distribution $\mu^* = \{\mu^*_n\}_{n \in \mathbb{N}}$ over the instances of CWM in P, with $m + 1$ candidates $p, c_1, \ldots, c_m$, where each μ^*_n is induced by the following sampling algorithm:

1. $\forall v \in T$: Randomly and independently choose $w(v) \in [0, 1]$ (up to $O(\log n)$ bits of precision).
2. $\forall i \in \{1, \ldots, m\}$: Randomly and independently choose $S[c_i] \in [(\alpha_1 - \alpha_2)W, \alpha_1 W]$ (up to $O(\log n)$ bits of precision).

We assume that $S[p] = 0$, i.e., all voters in S rank p last. This assumption is not a restriction. If it holds for a candidate c that $S[c] \leq S[p]$, then candidate c will surely lose, since the colluders all rank p first. Therefore, if $S[p] > 0$, we may simply normalize the scores by subtracting $S[p]$ from the scores of all candidates. This is equivalent to our assumption.

Remark 4. We believe that μ^* is the most natural distribution with respect to which coalitional manipulation in scoring protocols should be studied. Even if one disagrees with the exact definition of junta distribution, μ^* should satisfy many reasonable conditions one could produce.

We shall, of course, (presently) prove that the distribution possesses the properties of a junta distribution.

Proposition 4. *Let P be a sensitive scoring protocol. Then μ^* is a junta distribution for SCWM in P with $C = \{p, c_1, \ldots, c_m\}$, and $m = O(1)$.*

Proof. We first observe that the dichotomy and symmetry conditions are obviously satisfied.

The proof of the hardness property relies on the reduction from Partition in Proposition 2. The reduction generates instances x of CWM in P with 3 candidates, where $W = 4(\alpha_1 + \alpha_2)K$, and

$$\begin{aligned}S[a] &= S[b]\\ &= (K(4\alpha_1 - 2\alpha_2) - 1)(\alpha_1 + \alpha_2)\\ &= (\alpha_1 - \alpha_2/2)W - (\alpha_1 + \alpha_2),\end{aligned}$$

for some K that originates in the Partition instance. These instances satisfy $(\alpha_1 - \alpha_2)W \leq S[a], S[b] \leq \alpha_1 W$. It follows that $\mu^*(x) > 0$ (after scaling down the weights).[3]

We now prove μ^* has the balance property. If for all i, $S[c_i] > (\alpha_1 - \alpha_2/m)W$, then clearly there is no manipulation, since at least $\alpha_2 W$ points are given by the voters in T to the undesirable candidates $c_1, \ldots, c_m$. This happens with probability at least $\frac{1}{m^m}$.

On the other hand, consider the situation where for all i,

$$S[c_i] < (\alpha_1 - \frac{m^2 - 1}{m^2}\alpha_2)W; \qquad (3)$$

this occurs with probability at least $\frac{1}{(m^2)^m}$. Intuitively, if the colluders could distribute the votes in T in such a way that each undesirable candidate is ranked last in exactly $1/m$-fraction of the votes, this would be a successful manipulation: each undesirable candidate would gain at most an additional $\frac{m-1}{m}\alpha_2 W$ points. Unfortunately, this is usually not the case, but the following condition is sufficient for a successful manipulation (assuming condition (3) holds). Partition the voters in T to m disjoint subsets $p_1, \ldots, p_i$ (w.l.o.g. of size n/m), and denote by W_{p_i} the total weight of the votes in p_i. The condition is that for all $i \in \{1, \ldots, m\}$:

$$(1 - 1/m) \cdot 1/2 \cdot n/m \leq W_{p_i} \leq (1 + 1/m) \cdot 1/2 \cdot n/m. \quad (4)$$

This condition is sufficient, because if the voters in p_i all rank c_i last, the fraction of the votes in T which gives c_i points is at most:

$$\frac{(m-1)(1 + 1/m)}{(m-1)(1 + 1/m) + 1 - 1/m} = \frac{m^2 - 1}{m^2 + m - 2}.$$

Hence the number of points c_i gains from the colluders is at most:

$$\frac{m^2 - 1}{m^2 + m - 2}\alpha_2 \leq \frac{m^2 - 1}{m^2}\alpha_2 < \alpha_1 W - S[c_i].$$

Furthermore, by Lemma 1 and the fact that the expected total weight of n/m votes is $1/2 \cdot n/m$, the probability that condition (4) holds is at least $1 - 2e^{-\frac{2n}{m^3}}$. Since m is a constant, this probability is larger than $1/2$ for a large enough n.

[3] It seems the reduction can be generalized for a larger number of candidates. The hard instances are the ones where all undesirable candidates but two have approximately $(\alpha_1 - \alpha_2)W$ initial points, and two problematic candidates have approximately $(\alpha_1 - \alpha_m/2)W$ points. These instances have a positive probability under μ^*.

Algorithm 1 Decides SCWM

```
procedure GREEDY($S, \vec{w}, p$)
    for all $c \in C$ do                                  ▷ Initialization
        $S_0[c] \leftarrow S[c]$
    end for
    for $i = 1$ to $n$ do                                 ▷ All voters in $T$
        Let $j_1, j_2, \ldots, j_m$ s.t. $\forall l,\ S_{i-1}[c_{j_{l-1}}] \le S_{i-1}[c_{j_l}]$
        Voter $t_i$ votes $p \succ c_{j_1} \succ c_{j_2} \succ \ldots \succ c_{j_m}$
        for $l = 1$ to $m$ do                             ▷ Update score
            $S_i[c_{j_l}] \leftarrow S_{i-1}[c_{j_l}] + w(t_i)\alpha_{l+1}$
        end for
        $S_i[p] \leftarrow S_{i-1}[p] + w(t_i)\alpha_1$
    end for
    if $\mathrm{argmax}_{c \in C} S_n[c] = \{p\}$ then        ▷ $p$ wins
        return true
    else
        return false
    end if
end procedure
```

Finally, it can easily be seen that μ^* has the refinement property: if all colluders rank p first and candidate c second, then p gets $\alpha_1 W$ points, and c gets $\alpha_2 W + S[c]$ points. But $S[c] \ge (\alpha_1 - \alpha_2)W$, and thus p surely loses. □

4.2 A Heuristic Polynomial Time Algorithm

We now present our algorithm GREEDY for SCWM, given as Algorithm 1. $\vec{w}$ denotes the vector of the weights of voters in $T = \{t_1, \ldots, t_n\}$.

The voters in T, according to some order, each rank p first, and the rest of the candidates by their current score: the candidate with the lowest current score is ranked highest. GREEDY accepts if and only if p wins this election.

This algorithm, designed specifically for scoring protocols, is a realization of an abstract greedy algorithm: at each stage, voter t_i ranks the undesirable candidates in an order that minimizes the highest score that any undesirable candidate obtains after the current vote. If there is a tie between several permutations, the voter chooses the option such that the second highest score is as low as possible, etc. In any case, every colluder always ranks p first.

REMARK 5. This abstract scheme might also be appropriate for protocols such as Maximin and Copeland. Similarly to scoring protocols, in these two protocols the colluders are always better off by ranking p first. In addition, the abstract greedy algorithm can be applied to Maximin and Copeland since the result of an election is based on the score each candidate has (unlike STV, for example).

In the following lemmas, a *stage* in the execution of the algorithm is an iteration of the for loop.

LEMMA 2. *If there exists a stage i_0 during the execution of* GREEDY, *and two candidates $a, b \neq p$, such that*

$$|S_{i_0}[a] - S_{i_0}[b]| \le \alpha_2, \tag{5}$$

then for all $i \ge i_0$ it holds that $|S_i[a] - S_i[b]| \le \alpha_2$.

PROOF. The proof is by induction on i. The base of the induction is given by equation (5). Assume that $|S_i[a] - S_i[b]| \le \alpha_2$, and without loss of generality: $S_i[a] \ge S_i[b]$. By the algorithm, voter t_{i+1} ranks b higher than a, and therefore:

$$\mathrm{S}_{i+1}[b] - S_{i+1}[a] \ge -\alpha_2. \tag{6}$$

Since p is always ranked first, and the weight of each vote is at most 1, b gains at most α_2 points. Therefore:

$$S_{i+1}[b] - S_{i+1}[a] \le \alpha_2. \tag{7}$$

Combining equations (6) and (7) completes the proof. □

LEMMA 3. *Let $p \neq a, b \in C$, and suppose that there exists a stage i_0 such that $S_{i_0}[a] \ge S_{i_0}[b]$, and a stage $i_1 \ge i_0$ such that $S_{i_1}[b] \ge S_{i_1}[a]$. Then for all $i \ge i_1$ it holds that $|S_i[a] - S_i[b]| \le \alpha_2$.*

PROOF. Assume that there exists a stage i_0 such that $S_{i_0}[a] \ge S_{i_0}[b]$, and a stage $i_1 \ge i_0$ such that $S_{i_1}[b] \ge S_{i_1}[a]$; w.l.o.g. $i_1 > i_0$ (otherwise at stage i_0 it holds that $S_{i_0}[b] = S_{i_0}[a]$, and then we finish by Lemma 2). Then there must be a stage i_2 such that $i_0 \le i_2 < i_1$ and $S_{i_2}[a] \ge S_{i_2}[b]$ but $S_{i_2+1}[b] \ge S_{i_2+1}[a]$. Since the weight of each vote is at most 1, b gains at most α_2 points by voter t_{i_2+1}. Hence the conditions of Lemma 2 hold for stage i_2, which implies that for all $i \ge i_2$: $|S_i[a] - S_i[b]| \le \alpha_2$. In particular, $i_1 \ge i_2$. □

LEMMA 4. *Let P be a sensitive scoring protocol, and assume* GREEDY *errs on an instance of SCWM in P which has a successful manipulation. Then there is $d \in \{2, 3, \ldots, m\}$, and a subset of candidates $D = \{c_{j_1}, \ldots, c_{j_d}\}$, such that:*

$$\begin{aligned}&\sum_{i=1}^{d}(\alpha_1 W - S[c_{j_i}]) - \sum_{i=1}^{d-1}(i \cdot \alpha_2) \le W \sum_{i=1}^{d} \alpha_{m+2-i}\\ &\le \sum_{i=1}^{d}(\alpha_1 W - S[c_{j_i}]).\end{aligned} \tag{8}$$

PROOF. For the right inequality, for any d candidates, even if all voters in T rank them last in every vote, the total points distributed among them is $W \sum_{i=1}^{d} \alpha_{m+2-i}$. If this inequality does not hold, there must be some candidate c_i that gains at least $\alpha_1 W - S[c_i]$ points from the colluders, implying that this candidate has at least $\alpha_1 W$ points. However, p also has at most $\alpha_1 W$ points, and we assumed that there is a successful manipulation — a contradiction.

For the left inequality, assume the algorithm erred. Then at some stage i_0, there is a candidate c_{j_0} who has a total of at least $\alpha_1 W$ points (w.l.o.g. only one candidate passes this threshold simultaneously). Denote $T_0 = \{t_1, t_2, \ldots, t_{i_0}\}$, and let W_{T_0} be the total weight of the voters in T_0. Voter t_{i_0} did not rank c_{j_0} last, since $\alpha_{m+1} = 0$, and thus ranking a candidate last gives it no points. We have that there is another candidate c_{j_1}, such that: $S_{i_0-1}[c_{j_1}] \ge S_{i_0-1}[c_{j_0}]$. By Lemma 3, $S_{i_0}[c_{j_0}] - S_{i_0}[c_{j_1}] \le \alpha_2$, and thus $S_{i_0}[c_{j_1}] \ge \alpha_1 W - \alpha_2$. If these candidates were not always ranked last by the voters of T_0, there must be another candidate c_{j_2} who was ranked strictly higher by some voter in T_0, w.l.o.g. higher than c_{j_1}. Therefore, we have from Lemma 3 that: $S_{i_0}[c_{j_1}] - S_{i_0}[c_{j_2}] \le \alpha_2$, and so c_{j_2} has a total of at least $\alpha_1 W - 2\alpha_2$ points. By inductively continuing this reasoning, we obtain a subset D of d candidates (possibly $d = m$), who were always ranked in the d last places by the voters in T_0, and for the l'th candidate it holds that: $S_{i_0}[c_{j_l}] \ge \alpha_1 W - (l-1)\alpha_2$. The total points gained by this l'th candidate until stage i_0 must be at least $\alpha_1 W - (l-1)\alpha_2 - S[c_{j_l}]$.

Since the total points distributed by the voters in T_0 to the d last candidates is $W_{T_0}\sum_{i=1}^{d}\alpha_{m+2-i}$, we have:

$$\begin{aligned}\sum_{i=1}^{d}(\alpha_1 W - S[c_{j_i}]) - \sum_{i=1}^{d-1}(i\cdot\alpha_2) &\le W_{T_0}\sum_{i=1}^{d}\alpha_{m+2-i}\\ &\le W\sum_{i=1}^{d}\alpha_{m+2-i}.\end{aligned}$$

□

Lemma 5. *Let M be SCWM in a sensitive scoring protocol P with $C = \{p, c_1, \ldots, c_m\}$, m=O(1). Then* Greedy *is a deterministic heuristic polynomial time algorithm for $\langle M, \mu^*\rangle$.*

Proof. It is obvious that if the given instance has no successful manipulation, then the greedy algorithm would indeed answer that there is no manipulation, since the algorithm is constructive (it actually selects specific votes for the colluders).

We wish to bound the probability that there is a manipulation and the algorithm erred. By Lemma 4, a necessary condition for this to occur is as specified in equation (8), or equivalently:

$$\begin{aligned}&W\sum_{i=1}^{d}\alpha_1 - W\sum_{i=1}^{d}\alpha_{m+2-i} - \frac{d(d-1)}{2}\alpha_2 \le \sum_{i=1}^{d}S[c_{j_i}]\\ &\le W\sum_{i=1}^{d}\alpha_1 - W\sum_{i=1}^{d}\alpha_{m+2-i}.\end{aligned} \tag{9}$$

In this case the algorithm may err; but what is the probability of equation (9) holding? Fix a subset D of size $d \in \{2, \ldots, m\}$. $\sum_{i=1}^{d} S[c_{j_i}]$ is a random variable that takes values in $[d(\alpha_1-\alpha_2)W, d\alpha_1 W]$. By fixing values for the $S[c_{j_i}]$, we have that the probability of $\sum_{i=1}^{d} S[c_{j_i}]$ taking values in some interval $[a, b]$ is at most the chance of $S[c_{j_d}]$ taking a value in an interval of size $b - a$, which is at most $\frac{b-a}{\alpha_1 W-(\alpha_1-\alpha_2)W}$, since $S[c_{j_d}]$ is uniformly distributed. By Lemma 1, $W < n/4$ with probability at most $\epsilon(n) = e^{-\frac{n}{8}}$. On the other hand, if $W \ge n/4$, then (9) holds for D with probability at most

$$\frac{\frac{d(d-1)}{2}\alpha_2}{\alpha_1 W - (\alpha_1-\alpha_2)W} = \frac{d(d-1)}{2W} \le \frac{2d(d-1)}{n} = \frac{1}{p^D(n)},$$

for some polynomial p^D. We complete the proof by showing that equation (1) holds:

$$\begin{aligned}&\Pr_{x\sim\mu_n^*}[\textsc{Greedy}(x) \ne M(x)]\\ &\le \Pr[W \ge n/4 \wedge (\exists D \subset C \ s.t.\ |D| \ge 2 \wedge (9)\text{ holds})]\\ &\quad + \Pr[W < n/4]\\ &\le \sum_{D\subset C:|D|\ge 2}\frac{1}{p^D(n)} + \epsilon(n)\\ &\le \frac{1}{\text{poly}\, n}\end{aligned}$$

The last inequality follows from the assumption that $m = O(1)$. □

Clearly, Theorem 1 directly follows.

Algorithm 2 Decides UVWM

```
procedure Sample-and-Manipulate($\vec{w}, \nu, p, r$)
    for all permutations of the $m+1$ candidates do
        $\pi \leftarrow$ next permutation
        $\nu^* \leftarrow$ the manipulator votes $\pi$
            ▷ others' votes are always distributed w.r.t. $\nu$
        if Sample($\vec{w}$, $\nu^*$, $p$, $r$) then
            return true
        end if
    end for
    return false
end procedure
```

5. SUSCEPTIBILITY TO UVWM

In this section we shall prove:

Theorem 5. *Let P be a voting protocol such that there exists a junta distribution μ^P over the instances of UVWM in P, with the following property: r is uniformly distributed in $[0, 1]$. Then P, with candidates $C = \{p, c_1, \ldots, c_m\}$, $m = O(1)$, is susceptible to UVWM.*

The existence of a junta distribution with r uniformly distributed is a very weak requirement (it is even quite natural to have r uniformly distributed). In fact, the following claim is very likely to be true:

Conjecture 6. *Let P be a voting protocol. Then there exists a junta distribution μ^P over the instances of UVWM in P, with r uniformly distributed in $[0, 1]$.*

If this conjecture is indeed true, we have that all voting protocols are susceptible to UVWM. If for some reason the conjecture is not true with respect to our definition of junta distributions, then perhaps the definition is too restrictive and should be modified accordingly.

To prove Theorem 5, we require a procedure named Sample, which decides UVWE. Sample samples the given distribution on the votes n^3 times, and calculates the winner of the election each time. If p won more than an r-fraction of the elections then the procedure accepts, otherwise it rejects. We omit the details of the procedure, as well as the proof of the following lemma, due to lack of space.

Lemma 6. *Let P be a voting protocol, and E be UVWE in P with $C = \{p, c_1, \ldots, c_m\}$. Furthermore, let μ be a distribution over the instances of E, with r uniformly distributed in $[0, 1]$. Then there exists N such that for all $n \ge N$:*

$$\Pr_{x\sim\mu_n}[\textsc{Sample}(x) \ne E(x)] \le \frac{1}{\text{poly}n}.$$

We now present an algorithm, Sample-and-Manipulate that decides UVWM; it is given as Algorithm 2. Here, $\vec{w}$ denotes the weights of all voters including the manipulator, and ν is the given distribution over the others' votes.

Given an instance of UVWM, Sample-and-Manipulate generates $(m+1)!$ instances of the UVWE problem, one for each of the manipulator's possible votes, and executes Sample on each instance. Sample-and-Manipulate accepts if and only if Sample accepts one of the instances.

Lemma 7. *Let P be a voting protocol, and M be UVWM in P with $C = \{p, c_1, \ldots, c_m\}$, $m = O(1)$. Furthermore,* *let μ be a distribution over the instances of UVWM, with r uniformly distributed in* $[0, 1]$. *It holds that* Sample-and-Manipulate *is a probabilistic heuristic polynomial time algorithm for* $\langle M, \mu \rangle$.

Proof. For each independent call to Sample, the chance of failure is inverse-polynomial. By applying the union bound we have that the probability of Sample failing on any of the $(m+1)!$ invocations is at most $(m+1)!\frac{1}{\text{poly}n}$, which is still inverse-polynomial since m is constant. The lemma now follows from the fact that there is a manipulation if and only if there is a permutation of candidates, such that if the manipulator votes according to this permutation, the chance of p winning is greater than r.

Notice that Sample-and-Manipulate is indeed polynomial by the fact that $m = O(1)$, and we assumed that the given distribution over the votes can be sampled in polynomial time. □

6. FUTURE RESEARCH

The issue of resistance of mechanisms to manipulation is important, particularly in the context of voting protocols. Most results on this issue use $\mathcal{NP}$-hardness as the complexity measure. One of this paper's main contributions has been introducing tools that can be utilized in showing that manipulating mechanisms is *easy* in the average case. We were concerned with the likely case of coalitional manipulation, and showed that sensitive scoring protocols are susceptible to such manipulation when the number of candidates is constant.

These results suggest that scoring protocols cannot be safely employed. More importantly, this paper should be seen as a starting point for studying the average case complexity of other types of manipulations, in other protocols. In addition, the definitions in Section 3 are deliberately general, and can be applied to manipulations of mechanisms that are not voting mechanisms. One such mechanism of which we are aware, whose manipulation is $\mathcal{NP}$-hard, is presented in [1, 2].

There is still room for debate as to the exact definition of a junta distribution, especially if Conjecture 6 turns out to be false. It may also be the case that there are "unconvincing" distributions that satisfy all of the (current) conditions of a junta distribution. It might prove especially fruitful to show that a heuristic polynomial time algorithm with respect to a junta distribution also has the same property with respect to some easy distributions, such as the uniform distribution.

An issue of great importance is coming up with natural criteria to decide when a manipulation problem is *hard* in the average-case. The traditional definition of average-case completeness is very difficult to work with in general; is there a satisfying definition that applies specifically to the case of manipulations? Once the subject is more fully understood, this understanding can be used to design mechanisms that are hard to manipulate in the average-case.

7. ACKNOWLEDGMENT

This work was partially supported by grant #039-7582 from the Israel Science Foundation.

8. REFERENCES

[1] Y. Bachrach and J. S. Rosenschein. Achieving allocatively-efficient and strongly budget-balanced mechanisms in the network flow domain for bounded-rational agents. In *The Nineteenth International Joint Conference on Artificial Intelligence*, pages 1653–1654, Edinburgh, Scotland, August 2005.

[2] Y. Bachrach and J. S. Rosenschein. Achieving allocatively-efficient and strongly budget-balanced mechanisms in the network flow domain for bounded-rational agents. The Seventh International Workshop on Agent-Mediated Electronic Commerce (AMEC 2005), Utrecht, The Netherlands, July 2005.

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

[4] V. Conitzer, J. Lang, and T. Sandholm. How many candidates are needed to make elections hard to manipulate? In *Proceedings of the International Conference on Theoretical Aspects of Reasoning about Knowledge*, pages 201–214, Bloomington, Indiana, 2003.

[5] V. Conitzer and T. Sandholm. Complexity of manipulating elections with few candidates. In *Proceedings of the National Conference on Artificial Intelligence*, pages 314–319, Edmonton, Canada, July 2002.

[6] V. Conitzer and T. Sandholm. Universal voting protocol tweaks to make manipulation hard. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 781–788, Acapulco, Mexico, August 2003.

[7] E. Elkind and H. Lipmaa. Small coalitions cannot manipulate voting. In *International Conference on Financial Cryptography*, Lecture Notes in Computer Science. Springer-Verlag, Roseau, The Commonwealth of Dominica, 2005.

[8] A. Gibbard. Manipulation of voting schemes. *Econometrica*, 41:587–602, 1973.

[9] E. Hemaspaandra and L. A. Hemaspaandra. Dichotomy for voting systems. University of Rochester Department of Computer Science Technical Report 861, 2005.

[10] M. Satterthwaite. Strategy-proofness and Arrow's conditions: Existence and correspondence theorems for voting procedures and social welfare functions. *Journal of Economic Theory*, 10:187–217, 1975.

[11] L. Trevisan. Lecture notes on computational complexity. Available from http://www.cs.berkeley.edu/˜luca/notes/complexitynotes02.pdf, 2002. Lecture 12.