Journal of Artificial Intelligence Research 28 (2007) 157–181 Submitted 08/06; published 02/07

Junta Distributions and the Average-Case Complexity of Manipulating Elections

Ariel D. Procaccia ARIELPRO@CS.HUJI.AC.IL
Jeffrey S. Rosenschein JEFF@CS.HUJI.AC.IL
School of Engineering and Computer Science,
The Hebrew University of Jerusalem,
Jerusalem 91904, Israel

Abstract

Encouraging voters to truthfully reveal their preferences in an election has long been an important issue. Recently, computational complexity has been suggested as a means of precluding strategic behavior. Previous studies have shown that some voting protocols are hard to manipulate, but 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 some new concepts. In particular, we consider elections distributed with respect to *junta distributions*, which concentrate on hard instances. We use our techniques to prove that scoring protocols are susceptible to manipulation by coalitions, when the number of candidates is constant.

1. Introduction

Multiagent environments are often inhabited by heterogeneous, selfish agents, continually interacting but sharing few common goals. In such settings, agents may have diverse — or even conflicting — preferences. Therefore, reaching consensus among agents has long been an important issue.

A general, well-studied and well-understood 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 in the election can be beliefs, plans (Ephrati & Rosenschein, 1997), schedules (Haynes, Sen, Arora, & Nadella, 1997), or indeed many other less obvious entities, such as movies (Ghosh, Mundhe, Hernandez, & Sen, 1999). Applications of voting, in place of other methods, are motivated by theoretical guarantees provided by various voting protocols. For instance, Ghosh et al. (1999) present a movie recommender system that relies on voting, and makes use of voting properties to generate convincing explanations for different recommendations.

There is, however, an obstacle that has always plagued voting theory, and social choice theory in general: strategic behavior on the part of voters. In our setting, a self-interested agent may reveal its preferences untruthfully, if it believes this would make the final outcome of the elections more favorable for it. Manipulation is generally regarded as a problem, since it makes the actual ballot into a complex game, where the voters react and counter-react to the strategies of others. Not only does this require a larger investment of (computational) resources by voters, it may result in a socially undesirable alternative being chosen.

The celebrated Gibbard-Satterthwaite Theorem (Gibbard, 1973; Satterthwaite, 1975) establishes that in any deterministic voting protocol that is non-dictatorial,[1] there are elections where an agent is better off by voting untruthfully. Consequently, it is not possible to design a nonmanipulable voting system so as to guarantee that voters act honestly.

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. Indeed, it has been demonstrated that several voting protocols are $\mathcal{NP}$-hard to manipulate by a single voter (Bartholdi, Tovey, & Trick, 1989a; Bartholdi & Orlin, 1991). Hereinafter we mainly focus our attention on a setting in which multiple manipulators collude in order to achieve a certain outcome. In this setting, manipulation is even harder: it is known that the coalitional manipulation problem is $\mathcal{NP}$-hard in numerous voting protocols, even when the number of candidates is constant.

These results suggest that computational complexity may be the cure to the malady called "Manipulation". In Computer Science, though, the notion of hardness is usually considered in the sense of worst-case complexity. Indeed, most results on the complexity of manipulation use $\mathcal{NP}$-hardness as the complexity measure. Therefore, it could still be the case that most instances of the problem are easy to manipulate. To put it differently, a strategic voter may *usually* succeed in finding a beneficial manipulation, and do so efficiently, even when the problem is hard in the worst-case. If so, the truly significant issue is the *average-case* complexity of manipulations.

Sadly, so far all attempts to design a voting protocol that is resistant to manipulations in the average-case have failed. This suggests that the manipulation problem is inherently easy in the average-case — and pushes us to analytically support this claim: we must characterize settings and protocols that can *easily* be manipulated in the average-case.

A relatively little-known theory of average case complexity exists (Trevisan, 2002); that theory introduces the concept of distributional problems, and defines what a reduction between distributional problems is. It is also known that there are average-case complete problems. However, the goal of the 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*.

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 our 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 analytically establish the following result: an important family of voting protocols, called *scoring* protocols, is susceptible to coalitional manipulation

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

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^*. The significance of this result stems from the fact that sensitive scoring protocols are $\mathcal{NP}$-hard to manipulate, *even* when the number of candidates is constant. We support our claim that junta distributions provide a good benchmark by proving that Greedy also usually succeeds with respect to the uniform distribution.

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.

The paper proceeds as follows: in Section 2, we outline some important voting protocols, and 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. In Section 6 we survey related work. Finally, in Section 7, we present our 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 two useful lemmas from probability theory.

2.1 Elections and Manipulations

An election consists of a set $C = \{c_1, c_2, \ldots\}$ of candidates and a set $V = \{v_1, v_2, \ldots, \}$ of 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 the number of candidates is constant, so the complexity results are in terms of the number of voters.

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 $\vec{\alpha} = \langle \alpha_1, \alpha_2, \ldots, \alpha_{|C|} \rangle$, such that $\alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_{|C|}$ 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:* $\vec{\alpha} = \langle |C| - 1, |C| - 2, \ldots, 1, 0 \rangle$.

- *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 that are exponential in n can be scaled down to rational weights in the segment $[0, 1]$, represented using $O(n)$ bits.

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 $\vec{\alpha} = \langle \alpha_1, \alpha_2, \ldots, \alpha_{|C|} \rangle$. We say that P is *sensitive* iff $\alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_{|C|-1} > \alpha_{|C|} = 0$ (notice the strict inequality on the right).

In particular, Borda and Veto are sensitive scoring protocols.

Remark 2. Generally, from any scoring protocol with $\alpha_{|C|-1} > \alpha_{|C|}$, an equivalent sensitive scoring protocol can be obtained by subtracting $\alpha_{|C|}$ 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 $\alpha_{|C|} = 0$, then $\alpha_{|C|-1} = 0$. In this case, for three candidates it is equivalent to the plurality protocol, for which all interesting formulations of the manipulation problem are tractable even in the worst-case. Therefore, it is sufficient to restrict our results to sensitive scoring protocols.

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 (IM) 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.

2. This is a standard assumption, also made, for example, in the work of Conitzer and Sandholm (2002), and Conitzer, Lang, and Sandholm (2003).

Bartholdi and Orlin (1991) 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 well-known voting schemes, and hence will not be studied here.

In the lion's share of this paper, we consider the coalitional manipulation setting. In this scenario, the set V of voters is partitioned into two subsets: the set $V_1 = \{v_1, \ldots, v_n\}$ of manipulative, or untruthful, voters; and the set $V_2 = \{v_{n+1}, \ldots, v_{n+N}\}$ of nonmanipulative voters. The set of candidates is $C = \{c_1, \ldots, c_m, p\}$. The manipulators' goal is to make the distinguished candidate p win the election, by coordinating their rankings of candidates. In the CWM and SCWM problems, the manipulators have full knowledge of the nonmanipulators' votes.

Definition 3. In the Coalitional-Weighted-Manipulation (CWM) problem, we are given the set of voters $V = V_1 \uplus V_2$, the set of candidates C, the weights of all voters, and a preferred candidate $p \in C$. In addition, we are given the votes of the voters in V_2, and assume the manipulators are aware of these votes. We are asked whether it is possible for the manipulators in V_1 to cast their votes in a way that makes the preferred candidate p win the election.

We know (Conitzer & Sandholm, 2002; Conitzer et al., 2003) 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 the manipulators in V_1, and a preferred candidate p. We are asked whether it is possible for the manipulators in V_1 to cast their votes in a way that makes the preferred candidate p win the election.

$S[c]$ can be interpreted as c's total score from the votes in V_2. However, we do not require that there exist a combination of votes that actually induces $S[c]$ for all c.

Another setting that we shall shortly discuss (in Section 5) is the scenario where the manipulators are uncertain about the others' votes.

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 nonmanipulators' 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 for a protocol, then UVWE and UVWM are also $\mathcal{NP}$-hard for that protocol (Conitzer & Sandholm, 2002).

We make the assumption that the given distributions over the nonmanipulators' votes can be sampled in polynomial time. In other words, given a distribution over nonmanipulators' votes, it is possible to obtain a specific instance in polynomial time.

2.2 Probability Theory Tools

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)**.** *(Alon & Spencer, 1992) Let $X_1, \ldots, X_t$ be i.i.d. random variables such that $a \leq X_i \leq b$ and $\mathrm{E}[X_i] = \mu$. Then for any $\epsilon > 0$, it holds that:*

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

Another tool that we shall require is the Central Limit Theorem. For our purposes, it implies that the probability that a sum of random variables takes values in a very small segment is very small.

Lemma 2 (Central Limit Theorem)**.** *(Feller, 1968) Let $X_t, \ldots, X_t$ be independent continuous random variables with common density function, having expected value μ and variance σ^2. Then for $a < b$:*

$$\Pr\left[a < \frac{\sum_{i=1}^{t} X^i - t\mu}{\sqrt{t}\sigma} < b\right] \stackrel{t\to\infty}{\longrightarrow} \frac{1}{\sqrt{2\pi}} \int_a^b e^{-\frac{x^2}{2}} dx.$$

3. Our Approach

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 $\mu = \{\mu_n\}_{n\in\mathbb{N}}$, 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 that focuses on hard-to-manipulate instances. Ideally, we would like to ensure that if one manages to produce an algorithm that 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 $n \geq N$:

$$\frac{1}{c} \leq \ \mathrm{Pr}_{x \sim \mu_n}[M(x) = \text{“yes”}] \leq \ 1 - \frac{1}{c}.$$

3. Dichotomy: for all n and instances x such that $|x| = n$:

$$\mu_n(x) \geq 2^{-\mathrm{poly} n} \vee \mu_n(x) = 0.$$

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

4. Symmetry: Let v be a nonmanipulative voter, let $c_1, c_2 \neq p$ 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 manipulators 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 ensure that the junta distribution contains hard instances. Balance guarantees that a trivial algorithm that 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. (Trevisan, 2002) 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 Trevisan (2002) (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 of degree at least 1 and $N \in \mathbb{N}$ such that for all $n \geq N$:
$$\Pr_{x \sim \mu^n}[A(x) \neq M(x)] \leq \frac{1}{p(n)}. \tag{1}$$

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 of degree at least 1 and $N \in \mathbb{N}$ such that for all $n \geq N$:
$$\Pr_{x \sim \mu^n, s}[A(x) \neq M(x)] \leq \frac{1}{p(n)}. \tag{2}$$

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 used both to refer to the manipulation itself, and to 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. Formulation, Proof, and Justification of Main Result

Recall (Conitzer & Sandholm, 2002; Conitzer et al., 2003) 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. If $m = O(1)$ then P, with candidates $C = \{p, c_1, \ldots, c_m\}$, 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 V_1 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 Conitzer and Sandholm (2002) and Theorem 2 of Conitzer, Lang and Sandholm (2003), and justifies our focus on the family of sensitive scoring protocols. A stronger version of Proposition 1 has been independently proven by Hemaspaandra and Hemaspaandra (2005). Nevertheless, we include our proof, since it will be required in proving the hardness property of a junta distribution we shall design.

Proposition 1. *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 1. We reduce an arbitrary instance of Partition to the following CWM instance. There are 3 candidates, a, b, and p. In V_2, 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 V_1, for every k_i there is a vote of weight $2(\alpha_1 + \alpha_2)k_i$. Observe that from V_2, 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 V_1 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

$$(K(4\alpha_1 - 2\alpha_2) - 1)(\alpha_1 + \alpha_2) + 2K(\alpha_1 + \alpha_2)\alpha_2 = (\alpha_1 + \alpha_2)(4K\alpha_1 - 1)$$

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 V_1 vote either $p \succ a \succ b$ or $p \succ b \succ a$, because the manipulators 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 V_2. Therefore, a and b must gain less than $(2\alpha_2 K + 1)(\alpha_1 + \alpha_2)$ points from the voters in V_1. 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 into an instance of SCWM in the obvious way, we have:

Corollary 1. *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 V_1; P is a sensitive scoring protocol. We denote $|V_1| = n$: the size of V_1 is the size of the instance.

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

1. Fix a polynomial $q = q(n)$.
2. $\forall v \in T$: Randomly and independently choose $w(v) \in [0,1]$ (up to $O(n)$ bits of precision, i.e., in intervals of $1/2^{q(n)}$).
3. $\forall i \in \{1, \ldots, m\}$: Randomly and independently choose $S[c_i] \in [(\alpha_1 - \alpha_2)W, \alpha_1 W]$ (up to $O(n)$ bits of precision).

Remark 4. Although the distribution is in fact discrete — the weights, for example, are uniformly distributed in $\{0, 1/2^{q(n)}, 2/2^{q(n)}, 3/2^{q(n)}, \ldots, 1\}$ — we treat it below as continuous for the sake of clarity.

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 manipulators 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 5. 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 a 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 2. *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 symmetry is obviously satisfied, and dichotomy holds by Remark 4.

The proof of the hardness property relies on the reduction from Partition in Proposition 1. 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 that 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 V_1 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,

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

this occurs with probability at least $\frac{1}{(m^2)^m}$. Intuitively, if the manipulators could distribute their votes 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 manipulators to m disjoint subsets $P_1, \ldots, P_m$ (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 $i \in \{1, \ldots, m\}$:

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

This condition is sufficient, because if the voters in P_i all rank c_i last, the fraction of the votes in V_1 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 manipulators is at most:

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

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.

Finally, it can easily be seen that μ^* has the refinement property: if all manipulators rank p first and candidate c second, then p gets $\alpha_1 W$ points, and c gets $\alpha_2 W + S[c]$ points. But $S[c] \geq (\alpha_1 - \alpha_2)W$, 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 $V_1 = \{v_1, \ldots, v_n\}$.

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 V_1
    Let $j_1, j_2, \ldots, j_m$ s.t. $\forall l,\ S_{i-1}[c_{j_{l-1}}] \leq S_{i-1}[c_{j_l}]$
    Voter v_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

The voters in V_1, 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 v_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 among several permutations, the voter chooses the option such that the second highest score is as low as possible, etc. In any case, every manipulator always ranks p first.

Remark 6. This abstract scheme might also be appropriate for protocols such as Maximin and Copeland. Similarly to scoring protocols, in these two protocols the manipulators 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).

Remark 7. Greedy can be considered a generalization of the greedy algorithm given by Bartholdi et al. (1989a).

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

Lemma 3. *If there exists a stage i_0 during the execution of* Greedy*, and two candidates $a, b \neq p$, such that*

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

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

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

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

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

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

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

Lemma 4. *Let $p \neq a, b \in C$, and suppose that there exists a stage i_0 such that $S_{i_0}[a] \geq S_{i_0}[b]$, and a stage $i_1 \geq i_0$ such that $S_{i_1}[b] \geq S_{i_1}[a]$. Then for all $i \geq i_1$ it holds that $|S_i[a] - S_i[b]| \leq \alpha_2$.*

Proof. Assume that there exists a stage i_0 such that $S_{i_0}[a] \geq S_{i_0}[b]$, and a stage $i_1 \geq i_0$ such that $S_{i_1}[b] \geq S_{i_1}[a]$; 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 3). Then there must be a stage i_2 such that $i_0 \leq i_2 < i_1$ and $S_{i_2}[a] \geq S_{i_2}[b]$ but $S_{i_2+1}[b] \geq S_{i_2+1}[a]$. Since the weight of each vote is at most 1, b gains at most α_2 points by voter v_{i_2+1}. Hence the conditions of Lemma 3 hold for stage i_2, which implies that for all $i \geq i_2$: $|S_i[a] - S_i[b]| \leq \alpha_2$. In particular, $i_1 \geq i_2$. □

Lemma 5. *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:*

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

Proof. For the right inequality, for any d candidates, even if all voters in V_1 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 manipulators, 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 $V_1' = \{v_1, v_2, \ldots, v_{i_0}\}$, and let $W_{V_1'}$ be the total weight of the voters in V_1'. Voter v_{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: $S_{i_0-1}[c_{j_1}] \geq S_{i_0-1}[c_{j_0}]$. By Lemma 4, $S_{i_0}[c_{j_0}] - S_{i_0}[c_{j_1}] \leq \alpha_2$, and thus $S_{i_0}[c_{j_1}] \geq \alpha_1 W - \alpha_2$. If these candidates were not always ranked last by the voters of V_1', there must be another candidate c_{j_2} who was ranked strictly higher by some voter in V_1', w.l.o.g. higher than c_{j_1}. Therefore, we have from Lemma 4 that: $S_{i_0}[c_{j_1}] - S_{i_0}[c_{j_2}] \leq \alpha_2$, 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 V_1', and for the l'th candidate it holds that: $S_{i_0}[c_{j_l}] \geq \alpha_1 W - (l-1)\alpha_2$. 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 V_1' to the d last candidates is $W_{V_1'} \sum_{i=1}^{d} \alpha_{m+2-i}$, we have:

$$\sum_{i=1}^{d}(\alpha_1 W - S[c_{j_i}]) - \sum_{i=1}^{d-1}(i \cdot \alpha_2) \leq W_{V_1'} \sum_{i=1}^{d} \alpha_{m+2-i} \leq W \sum_{i=1}^{d} \alpha_{m+2-i}.$$

□

Lemma 6. *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 manipulators).

We wish to bound the probability that there is a manipulation and the algorithm erred. By Lemma 5, a necessary condition for this to occur is as specified in equation (8), or equivalently:

$$W \sum_{i=1}^{d} \alpha_1 - W \sum_{i=1}^{d} \alpha_{m+2-i} - \frac{d(d-1)}{2}\alpha_2 \leq \sum_{i=1}^{d} S[c_{j_i}] \leq W \sum_{i=1}^{d} \alpha_1 - W \sum_{i=1}^{d} \alpha_{m+2-i}. \quad (9)$$

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 conditioning on the values of $S[c_{j_i}]$, $i = 1, \ldots, d-1$, 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 $W \geq n/4$, then (9) holds for D with probability at most

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

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

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

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

Clearly, Theorem 1 directly follows.

4.3 Algorithm 1 and the Uniform Distribution

In the previous subsection we have seen that Algorithm 1 is a heuristic polynomial time algorithm with respect to our junta distribution μ^*. We have argued that this suggests that the algorithm also does well with respect to other distributions. In this subsection we support this claim by showing that Algorithm 1 is also a heuristic polynomial time algorithm with respect to the uniform distribution over instances of SCWM.

For the sake of consistency with previous results, we shall consider a uniform distribution over votes which may produce unfeasible ballots. Nevertheless, equivalent results can be obtained for feasible (discrete) distributions over votes. If so, in this subsection we assume each voter $v_i \in V_2$, where $|V_2| = N$, awards each candidate $c \in C$, including p, a score independently and uniformly distributed in $[0, \alpha_1]$. Further, we assume that the votes are unweighted; this does not limit the generality of our results, since we use lower bounds that depend only on the total weight of the manipulators in V_1 (where $|V_1| = n$) — the individual weights are of no consequence.

We distinguish between two cases in our results, depending on the ratio between the number of nonmanipulators N and the number of manipulators n:

1. $n/\sqrt{N} < 1/p(n)$ for some polynomial p of degree at least 1.

2. $n/\sqrt{N} > p(\log n)$ for some polynomial p of degree at least 1.

The middle ground which is not covered by the two cases remains an open problem. Before we tackle the first case, we require a lower bound of sorts on the probability that an instance of SCWM is very easy to decide. Since the manipulators in V_1 can award a candidate at most $\alpha_1 n$ points, the manipulators cannot make a candidate c beat another candidate c' if $S[c'] - S[c] > \alpha_1 n$. In particular, if for every two candidates c and c' it holds that $|S[c] - S[c']| > \alpha_1 n$, then the manipulators cannot affect the outcome of the election. Moreover, Algorithm 1 always decides such an instance correctly: if $S[p] < S[c]$ for some c, then the instance is a "no" instance, and in this case the algorithm never errs; and if $S[p] > S[c]$ for all c, then the instance is a "yes" instance, and any vote of the manipulators is sufficient to make p win. We have obtained the following Lemma:

Lemma 7. *Consider an instance of SCWM where for all $c, c' \in C$, $|S[c] - S[c']| > \alpha_1 n$. Then the instance is a "yes" instance iff $S[p] > S[c]$ for all candidates $c \neq p$, and the instance is correctly decided by Algorithm 1.*

This Lemma, together with the Central Limit Theorem, yields the first result.

Proposition 3. *Algorithm 1 is a heuristic polynomial time algorithm with respect to the uniform distribution over instances of SCWM which satisfy $n/\sqrt{N} < 1/p(n)$ for some polynomial $p(n)$ of degree at least 1.*

Proof. By Lemma 7, it is sufficient to bound from below the probability that for all $c, c' \in C$, $|S[c] - S[c']| > \alpha_1 N$.

$$\Pr[\forall c, c' \in C,\ |S[c] - S[c']| > \alpha_1 N] = 1 - \Pr[\exists c, c' \in C\ s.t.\ 0 \le S[c] - S[c'] \le \alpha_1 N].$$

By the union bound:

$$\Pr[\exists c, c' \in C \ s.t. \ 0 \leq S[c] - S[c'] \leq \alpha_1 n] \leq \sum_{c,c' \in C} \Pr[0 \leq S[c] - S[c'] \leq \alpha_1 n].$$

Fix $c, c' \in C$, and let X_i be $s^i[c] - s^i[c']$, where s^i is the score given to a candidate by voter $v_i \in V_2$, $i = n+1, \ldots, n+N$. The X_i are i.i.d. continuous random variables with expectation 0 and constant variance $\sigma^2 = 2 \cdot (\alpha_1)^2/12 = (\alpha_1)^2/6$. Therefore, we can apply Lemma 2:

$$\begin{aligned}
\Pr\left[0 \leq S[c] - S[c'] \leq \alpha_1 n\right] &= \Pr\left[0 \leq \sum_{i=n+1}^{n+N} X_i \leq \alpha_1 n\right] \\
&= \Pr\left[0 \leq \frac{\sum_{i=n+1}^{n+N} X^i}{\sqrt{N}\sigma} \leq \frac{\alpha_1 n}{\sqrt{N}\sigma}\right] \\
&\overset{N\to\infty}{\longrightarrow} \frac{1}{\sqrt{2\pi}} \int_0^{\frac{\alpha_1 n}{\sqrt{N}\sigma}} e^{-\frac{x^2}{2}} dx \\
&\leq \int_0^{\frac{\alpha_1 n}{\sqrt{N}\sigma}} 1 \ dx \\
&= \frac{\alpha_1 n}{\sqrt{N}\sigma} \\
&= O\left(\frac{n}{\sqrt{N}}\right).
\end{aligned}$$

By our assumption regarding the ratio of manipulators and nonmanipulators, this is inverse-polynomial in n. Rolling back, we obtain that the probability that the algorithm is correct is at least $1 - (m+1)m\frac{1}{p(n)}$, and the result follows from the fact that $m = O(1)$. $\square$

Moving on to the second case, we require the following lemma:

Lemma 8. *Let $\epsilon = \frac{\alpha_2}{2(m+1)}$, and consider an instance of SCWM where for all $c, c' \in C$, $|S[c] - S[c']| < \epsilon n$. Then this instance is a "yes" instance, and is correctly decided by Algorithm 1.*

Proof. Obviously, it is sufficient to prove that the algorithm constructively finds a successful vote that makes p win. Let $C' \subseteq C \setminus \{p\}$ be the set of undesirable candidates that had maximal score among the candidates in $C \setminus \{p\}$ at some stage during the execution of the algorithm. By the algorithm, at any stage some candidate from C' is ranked last by a voter in V_1, i.e., is given 0 points; the other candidates in C' receive at any stage at most α_2 points. Therefore, the total number of points the candidates in C' receive from the manipulators is at most $(d-1)\alpha_2 n$, where $|C'| = d$. Consequently, if $S^*[c]$ is the score of candidate c when the algorithm terminates,

$$\sum_{c \in C'} S^*[c] \leq \sum_{c \in C'} S[c] + (d-1)\alpha_2 n.$$

Let $c_0^* = \text{argmax}_{c \in C'} S[c]$, and $c_1^* = \text{argmax}_{c \in C'} S^*[c]$. By Lemma 4, when the algorithm terminates it holds that the scores of all candidates in C' are within α_2 of one another. Therefore:

$$S^*[c_1^*] \leq \sum_{c \in C'} S[c] + (d-1)\alpha_2 n - \sum_{c_1^* \neq c \in C'} S^*[c] \leq dS^*[c_0^*] + (d-1)\alpha_2 n - (d-1)(S^*[c_1^*] - \alpha_2).$$

Through some algebraic manipulations, we obtain:

$$S^*[c_1^*] \leq S[c_0^*] + n\left(\frac{d-1}{d}\alpha_2\right) + \frac{d-1}{d}\alpha_2 \leq S[c_0^*] + n\left(\frac{m}{m+1}\alpha_2\right) + \frac{m}{m+1}\alpha_2.$$

Now, we have that:

$$\begin{aligned} S^*[p] - S^*[c_1^*] &\geq (S[p] + \alpha_1 n) - \left(S[c_0^*] + \left(\frac{m}{m+1}\alpha_2\right) n + \frac{m}{m+1}\alpha_2\right) \\ &\geq \alpha_1 n - \frac{\alpha_2}{2(m+1)} n - \left(\frac{m}{m+1}\alpha_2\right) n - \frac{m}{m+1}\alpha_2 \\ &\geq \frac{\alpha_2}{2(m+1)} n - \frac{m}{m+1}\alpha_2 \\ &> 0. \end{aligned}$$

The second transition follows from the assumption that $S[p] \geq S[c_0^*] - \epsilon n$, the third transition from the fact that $\alpha_1 \geq \alpha_2$, and the last transition holds for a large enough n. $\square$

Proposition 4. *Algorithm 1 is a heuristic polynomial time algorithm with respect to the uniform distribution over instances of SCWM which satisfy $n/\sqrt{N} > p(\log n)$ for some polynomial p of degree at least 1.*

Proof. Let $\epsilon = \frac{\alpha_2}{2(m+1)}$. By Lemma 8, the probability that the algorithm does not err is at least:

$$\Pr[\forall c, c' \in C,\ |S[c] - S[c']| < \epsilon n] = 1 - \Pr[\exists c, c' \in C \ s.t.\ S[c] - S[c'] > \epsilon n].$$

By the union bound:

$$\Pr[\exists c, c' \in C \ s.t.\ S[c] - S[c'] > \epsilon n] \leq \sum_{c, c' \in C} \Pr[S[c] - S[c'] > \epsilon n].$$

As before, fix $c, c' \in C$, and let X_i be $s^i[c] - s^i[c']$, where s^i is the score given to a candidate by voter v_i. The X_i are i.i.d. random variables with expectation 0, which take values in $[-\alpha_1, \alpha_1]$. Applying Lemma 1 to these variables, we obtain:

$$\Pr[S[c] - S[c'] \geq \epsilon n] = \Pr[\frac{1}{N}\sum_{i=n+1}^{n+N} X_i \geq \mathbb{E}[X_i] + \frac{\epsilon n}{N}] \leq e^{-2N\frac{\left(\frac{\epsilon n}{N}\right)^2}{(2\alpha_1)^2}} = e^{-\epsilon' \frac{n^2}{N}},$$

where ϵ' is some constant. The result follows from the fact that m is constant and our assumption regarding the relation between n and N. $\square$

5. The Case of Uncertainty about Votes

So far we have dealt with a setting where an entire coalition of manipulators is trying to influence the outcome of the election, using complete knowledge of the nonmanipulators' votes. This section is a short aside, in which we discuss a setting where there is a single manipulator with uncertainty about others' votes. We shall prove:

Theorem 2. *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.*

Recall that in UVWM, we ask whether the manipulator can cast his vote so that p wins with probability greater than r. 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 1. Let P be a voting protocol for which UVWM is $\mathcal{NP}$-hard. 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. We also remark that similar results can be derived for destructive manipulations by analogous proofs.

To prove Theorem 2, we first present a helpful procedure, which decides UVWE. $\vec{w}$ denotes the vector of given weights, and ν is the given distribution over all the votes. The number of voters is $|V| = n$.

Sample($C = \{p, c_1, \ldots, c_m\}$, $\vec{w}$, ν, r)
count = 0
for $i = 1$ to n^3 **do**
    Sample the distribution ν over the votes
    Calculate the result of the election using the sampled votes
    if p won **then**
        count = count + 1
    end if
end for
if count$/n^3 > r$ **then**
    return 1
else
    return 0
end if

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.

Lemma 9. *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 $n \geq N$:*

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

Proof. Let $\{X_i\}_{i=1}^{n^3}$ be random variables, such that $X_i = 1$ if p won in the i'th iteration of the for loop, and $X_i = 0$ otherwise. Let r' be the probability that p wins in the given instance. By Lemma 1 and the union bound:

$$\Pr\left[\left|\frac{1}{n^3}\sum_{i=1}^{n^3} X_i - r'\right| \geq \frac{1}{n}\right] \leq 2e^{-2n^3\frac{1}{n^2}} = 2e^{-2n}.$$

We deduce that if $|r - r'| > \frac{1}{n}$, Sample will fail with an exponentially small probability. By the assumption that r is uniformly distributed, the probability that $|r - r'| \leq \frac{1}{n}$ is at most $2/n$. Thus, by the union bound it holds that:

$$\begin{aligned}\Pr_{x \sim \mu_n}[\textsc{Sample}(x) \neq E(x)] &\leq \Pr\left[|r - r'| \leq \frac{1}{n}\right] + \Pr\left[|r - r'| > \frac{1}{n} \wedge \textsc{Sample}(x) \neq E(x)\right] \\ &\leq 2/n + 2e^{-2n} \\ &\leq \frac{1}{\text{poly} n}.\end{aligned}$$

□

We now present an algorithm that decides UVWM. Here, $\vec{w}$ denotes the weights of all voters including the manipulator, and ν is the given distribution over the nonmanipulators' votes.

Sample-and-manipulate($C = \{p, c_1, \ldots, c_m\}$, $\vec{w}$, ν, r)

ans $= 0$
for $i = 1$ to $(m+1)!$ **do**
    π = next permutation of the $m+1$ candidates
    ν^* = the manipulator always votes π, others' votes are distributed with respect to ν
    if Sample(C, $\vec{w}$, ν^*, r) $= 1$ **then**
        ans $= 1$
    end if
end for
return ans

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 10. *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* $[0, 1]$. *Then* 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. Related Work

Computational aspects of voting have long been investigated. A pivotal issue is the problem of winner-determination: voting protocols designed to satisfy theoretical desiderata may be quite complex. Consequently, deciding who won an election governed by such protocols may be a computationally hard problem (Bartholdi, Tovey, & Trick, 1989b). Another concern is strategic behavior on the part of the officials conducting the election, who may add or remove voters and candidates from the slate. The computational complexity of strategically controlling an election has been analyzed by Bartholdi, Tovey and Trick (1992).

That said, the main issue with respect to strategic behavior in voting has always been manipulation by voters. There is a growing body of work which deals with the *worst-case* complexity of manipulating elections. A seminal paper is that of Bartholdi, Tovey and Trick (1989a); the authors suggested, for the first time, that computational complexity is an obstacle that strategic voters must overcome. Indeed, although it is shown that many voting protocols can be efficiently manipulated, it is nevertheless proven that there is a voting protocol, namely second-order Copeland, which is $\mathcal{NP}$-hard to manipulate. Bartholdi and Orlin (1991) have demonstrated that the prominent Single Transferable Vote (STV) protocol is $\mathcal{NP}$-hard to manipulate.

Even in voting protocols that are easy to manipulate, difficulty can be artificially introduced by adding a preround (Conitzer & Sandholm, 2003); the candidates are paired, and in each pairing of two candidates, the candidate that loses the pairwise election between the two is eliminated. Plurality, Borda and Maximin have been shown to be hard to manipulate when augmented with a preround. In more detail, these protocols are $\mathcal{NP}$-hard to manipulate when the scheduling of the preround precedes voting, $\#\mathcal{P}$-hard when voting precedes scheduling, and $\mathcal{PSPACE}$-hard when voting and scheduling are interleaved. Elkind and Lipmaa (2005a) induce hardness of manipulation using a more general approach. Hybrid voting protocols that are hard to manipulate are constructed by composing several base protocols; the base protocols may be individually easy to manipulate.

Another case where manipulation may be hard, in the worst-case, is when the election has multiple winners instead of a single winner, as is the case in elections to a parliament or an assembly. Procaccia, Rosenschein, and Zohar (2007) demonstrate that manipulation in Cumulative voting, a major protocol for multi-winner elections, is $\mathcal{NP}$-hard.

The coalitional manipulation problem, which has been the focus of this paper, has first been investigated by Conitzer and Sandholm (2002, 2003). In this setting, the computational problem is made more difficult by the fact that numerous manipulators must coordinate their strategy (and additionally, by the introduction of weighted voting). While the hardness results in the abovementioned papers relied on the number of candidates being unbounded, Conitzer and Sandholm present hardness results in the coalitional manipulation setting with a constant number of candidates, with respect to several central voting protocols.

Elkind and Lipmaa (2005b) extend the preround approach presented by Conitzer and Sandholm (2003) to the coalitional manipulation setting. In this context, they provide an early impossibility result regarding the average-case complexity of manipulations: the authors present a family of preference profiles in which a manipulator can always improve the outcome by voting strategically, regardless of the preround schedule. This result applies only when seeking to make manipulation hard by adding a preround. Further, one would usually not expect distributions over the instances of the coalitional manipulation problem to give this family of preference profiles significant probability, as it is extremely restricted.

A recent result regarding average-case complexity of manipulation, which complements our own, was presented by Conitzer and Sandholm (2006). The authors put forward two properties of instances of the coalitional manipulation problem, and demonstrate that any instance that satisfies both properties is easy to manipulate. The first property is that the instance satisfy a weaker form of monotonicity — this property seems very natural; the second property is that manipulators be able to make one of exactly two candidates win the election — and this property is much harder to accept. In order to justify the second property, the authors show that in many voting protocols the property usually holds, but only with respect to a specific family of distributions.

This result has two main shortcomings compared to ours. First, the arguments in favor of the second property mentioned above are empirical rather than analytical; second, the family of distributions considered is not "special" in any sense — which is not the case here. In other words, the family of distributions in question is *a priori* not especially hard to manipulate. On the other hand, their result has some advantages: unlike ours, it does not depend on the number of candidates being constant (although in all experiments the number of candidates and manipulators is extremely small compared to the number of voters), and (arguably) does not require significant restrictions on the voting rule.

7. Conclusions

To date, all results on the complexity of manipulation only considered the worst case. Although better than nothing, such results are a weak guarantee of resistance to manipulation. A truly worthy goal is to design a voting protocol that is hard to manipulate in the average case while being plausible from a social choice point of view, but so far all attempts have failed.

Motivated by this, we have presented a specific manipulation setting that is worst-case hard but average-case tractable. We have first prepared the ground for our average-case analysis by borrowing several concepts from the existing theory and introducing several new ideas. The key to our approach is junta distributions, which presumably concentrate on hard instances of the coalitional manipulation problem. We have considered a voting protocol to be susceptible to coalitional manipulation if there is an algorithm that almost always correctly decides the problem, when the instances are distributed with respect to a junta distribution.

Our main result states that sensitive scoring protocols are susceptible to coalitional manipulation when the number of candidates is constant, although they are hard to manipulate *even* when the number of candidates is constant.

7.1 Discussion

Our results, first and foremost, suggest that worst-case hardness is indeed not a strong enough barrier against manipulation. This motivates further research regarding average-case complexity of manipulations, at the expense of future investigations into worst-case complexity.

Moreover, in our view, our main result provides further evidence that a voting rule that is average-case hard to manipulate does not exist. At the very least, it suggests that scoring protocols cannot form the basis of a protocol which is usually strategy resistant. Nevertheless, this negative result can be circumvented in many ways.

First, it can be circumvented via the voting protocol. Scoring protocols are among the easiest voting systems to manipulate, as their structure is quite simple and they can be concisely represented. Other protocols, say STV, are inherently harder to deal with. In fact, recall that STV is worst-case hard to manipulate when there is only one manipulator (but an unbounded number of candidates) (Bartholdi & Orlin, 1991), whereas scoring protocols are most certainly not.

Second, it can be circumvented via the setting. Our results hold only when one contemplates coalitional manipulation with a constant number of candidates. A constant number of candidates is known to guarantee worst-case hardness, but it may be the case that allowing for a large number of candidates would make the difference with respect to average-case analysis.

Third, it can be circumvented via the distribution. Traditional average-case complexity theory deals with hardness of distributional problems; in other words, a specific distribution is considered. Junta distributions were chosen in a way that if one of them can usually be manipulated by an algorithm, presumably the same algorithm would be successful with other distributions. This view was supported by the results in Section 4.3, but at this point there are no strong theoretical guarantees, and it may certainly be true that there is a specific distribution over the instances of the manipulation problem which is average-case hard to manipulate, even when a scoring protocol is considered.

Section 4.3 deserves an aside. The lemmas established there show that, with respect to the uniform distribution, even a completely trivial algorithm can usually decide the coalitional manipulation problem: if the number of manipulators is small (less than the square root of the number of voters), the manipulators can rarely influence the outcome of the election; therefore, if p was elected by the nonmanipulators as well, it is usually correct to answer "yes", and if not, it is usually correct to answer "no". If the number of manipulators is large, it is usually correct to answer "yes" — there is a manipulation.

Recent preliminary results in this direction imply that this is true for several families of voting rules, and under a large variety of distributions. It is very important to note that such a simple algorithm *would not* work well with respect to the junta distribution μ^*.

7.2 Future Research

In our view, a central contribution of the paper is that it establishes a framework that can be used to study the average-case complexity of manipulations in other protocols, and even more generally, in other mechanisms. Indeed, voting is the most general method of preference aggregation, but the same issues are also relevant when one considers mechanisms in more specific settings. One such mechanism of which we are aware, whose manipulation is $\mathcal{NP}$-hard, is presented by Bachrach and Rosenschein (2006). All the definitions in Section 3 are sufficiently general to deal with different mechanisms for preference aggregation.

There is still room for debate as to the exact definition of a junta distribution. 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 *is guaranteed* to have 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 will surely shed more light on the great mystery: are there voting protocols that are usually hard to manipulate?

Acknowledgments

This work was partially supported by grant #898/05 from the Israel Science Foundation.

References

Alon, N., & Spencer, J. H. (1992). *The Probabilistic Method*. Wiley and Sons.

Bachrach, Y., & Rosenschein, J. S. (2006). Achieving allocatively-efficient and strongly budget-balanced mechanisms in the network flow domain for bounded-rational agents. In *The Seventh International Workshop on Agent-Mediated Electronic Commerce: Designing Mechanisms and Systems, Utrecht, The Netherlands (AMEC 2005)*, No. 3937 in Lecture Notes in Artificial Intelligence, pp. 71–84. Springer-Verlag, Berlin.

Bartholdi, J., & Orlin, J. (1991). Single transferable vote resists strategic voting. *Social Choice and Welfare*, *8*, 341–354.

Bartholdi, J., Tovey, C. A., & Trick, M. A. (1989a). The computational difficulty of manipulating an election. *Social Choice and Welfare*, *6*, 227–241.

Bartholdi, J., Tovey, C. A., & Trick, M. A. (1989b). Voting schemes for which it can be difficult to tell who won the election. *Social Choice and Welfare*, *6*, 157–165.

Bartholdi, J., Tovey, C. A., & Trick, M. A. (1992). How hard is it to control an election. *Mathematical and Computer Modelling*, *16*, 27–40.

Conitzer, V., Lang, J., & Sandholm, T. (2003). How many candidates are needed to make elections hard to manipulate?. In *Proceedings of the International Conference on Theoretical Aspects of Reasoning about Knowledge*, pp. 201–214.

Conitzer, V., & Sandholm, T. (2002). Complexity of manipulating elections with few candidates. In *Proceedings of the National Conference on Artificial Intelligence*, pp. 314–319.

Conitzer, V., & Sandholm, T. (2003). Universal voting protocol tweaks to make manipulation hard. In *Proceedings of the International Joint Conference on Artificial Intelligence*, pp. 781–788.

Conitzer, V., & Sandholm, T. (2006). Nonexistence of voting rules that are usually hard to manipulate. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence*, pp. 627–634.

Elkind, E., & Lipmaa, H. (2005a). Hybrid voting protocols and hardness of manipulation. In *16th Annual International Symposium on Algorithms and Computation*, Lecture Notes in Computer Science, pp. 206–215. Springer-Verlag.

Elkind, E., & Lipmaa, H. (2005b). Small coalitions cannot manipulate voting. In *International Conference on Financial Cryptography*, Lecture Notes in Computer Science. Springer-Verlag.

Ephrati, E., & Rosenschein, J. S. (1997). A heuristic technique for multiagent planning. *Annals of Mathematics and Artificial Intelligence*, *20*, 13–67.

Feller, W. (1968). *Introduction to Probability Theory and its Applications* (3rd edition)., Vol. 1, p. 254. John Wiley.

Ghosh, S., Mundhe, M., Hernandez, K., & Sen, S. (1999). Voting for movies: the anatomy of a recommender system. In *Proceedings of the Third Annual Conference on Autonomous Agents*, pp. 434–435.

Gibbard, A. (1973). Manipulation of voting schemes. *Econometrica*, *41*, 587–602.

Haynes, T., Sen, S., Arora, N., & Nadella, R. (1997). An automated meeting scheduling system that utilizes user preferences. In *Proceedings of the First International Conference on Autonomous Agents*, pp. 308–315.

Hemaspaandra, E., & Hemaspaandra, L. A. (2005). Dichotomy for voting systems. University of Rochester Department of Computer Science Technical Report 861.

Procaccia, A. D., Rosenschein, J. S., & Zohar, A. (2007). Multi-winner elections: Complexity of manipulation, control and winner-determination. In *The Twentieth International Joint Conference on Artificial Intelligence (IJCAI 2007)*, Hyderabad, India. To appear.

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

Trevisan, L. (2002). Lecture notes on computational complexity. Available from http://www.cs.berkeley.edu/~luca/notes/complexitynotes02.pdf. Lecture 12.