Journal of Artificial Intelligence Research 42 (2011) 1-29 Submitted 11/10; published 09/11

Where Are the Hard Manipulation Problems?

Toby Walsh TOBY.WALSH@NICTA.COM.AU
NICTA and UNSW
Sydney
Australia

Abstract

Voting is a simple mechanism to combine together the preferences of multiple agents. Unfortunately, agents may try to manipulate the result by mis-reporting their preferences. One barrier that might exist to such manipulation is computational complexity. In particular, it has been shown that it is NP-hard to compute how to manipulate a number of different voting rules. However, NP-hardness only bounds the worst-case complexity. Recent theoretical results suggest that manipulation may often be easy in practice. In this paper, we show that empirical studies are useful in improving our understanding of this issue. We consider two settings which represent the two types of complexity results that have been identified in this area: manipulation with unweighted votes by a single agent, and manipulation with weighted votes by a coalition of agents. In the first case, we consider Single Transferable Voting (STV), and in the second case, we consider veto voting. STV is one of the few voting rules used in practice where it is NP-hard to compute how a single agent can manipulate the result when votes are unweighted. It also appears one of the harder voting rules to manipulate since it involves multiple rounds. On the other hand, veto voting is one of the simplest representatives of voting rules where it is NP-hard to compute how a coalition of weighted agents can manipulate the result. In our experiments, we sample a number of distributions of votes including uniform, correlated and real world elections. In many of the elections in our experiments, it was easy to compute how to manipulate the result or to prove that manipulation was impossible. Even when we were able to identify a situation in which manipulation was hard to compute (e.g. when votes are highly correlated and the election is "hung"), we found that the computational difficulty of computing manipulations was somewhat precarious (e.g. with such "hung" elections, even a single uncorrelated voter was enough to make manipulation easy to compute).

1. Introduction

The Gibbard-Satterthwaite theorem proves that, under some weak assumptions like three or more candidates and the absence of a dictator, voting rules are manipulable (Gibbard, 1973; Satterthwaite, 1975). That is, it may pay for agents not to report their preferences truthfully. One appealing escape from this result was proposed by Bartholdi, Tovey and Trick (1989). Whilst a manipulation may exist, perhaps it is computationally too difficult to find. To illustrate this idea, they demonstrated that the second order Copeland rule is NP-hard to manipulate. Shortly after, Bartholdi and Orlin (1991) proved that the more well known Single Transferable Voting (STV) rule is NP-hard to manipulate. A whole subfield of social choice has since grown from this proposal, studying the computational complexity of manipulation and of the control of voting rules. Two good surveys have recently appeared that provide many references into the literature (Faliszewski, Hemaspaandra, & Hemaspaandra, 2010; Faliszewski & Procaccia, 2010). Computational complexity results about the manipulation of voting rules typically vary along five different dimensions.

Weighted or unweighted votes: Are the votes weighted or unweighted? Although many elections involve unweighted votes, weighted votes are used in a number of real-world settings like shareholder meetings, and elected assemblies. Weights are also useful in multi-agent systems where we have different types of agents. Weights are interesting from a computational perspective for at least two reasons. First, weights can increase computational complexity. For example, computing how to manipulate the veto rule is polynomial with unweighted votes but NP-hard with weighted votes (Conitzer, Sandholm, & Lang, 2007). Second, the weighted case informs us about the unweighted case when we have probabilistic information about the votes. For instance, if it is NP-hard to compute if an election can be manipulated with weighted votes, then it is NP-hard to compute the probability of a candidate winning when there is uncertainty about how the unweighted votes have been cast (Conitzer & Sandholm, 2002a).

Bounded or unbounded number of candidates: Do we have a (small) fixed number of candidates? Or is the number of candidates allowed to grow? For example, with unweighted votes, computing a manipulation of the STV rule is polynomial if we bound the number of candidates and only NP-hard when the number of candidates is allowed to grow with problem size (Bartholdi & Orlin, 1991). Indeed, with unweighted votes and a bounded number of candidates, it is polynomial to compute how to manipulate most voting rules (Conitzer et al., 2007). On the other hand, with weighted votes, it is NP-hard to compute how to manipulate many voting rules with a bounded number of candidates. For example, it is NP-hard to compute a manipulation with the veto rule and 3 or more candidates (Conitzer et al., 2007).

One manipulator or a coalition of manipulators: Is a single agent trying to manipulate the results or is a coalition of agents acting together? A single agent is unlikely to be able to change the outcome of many elections. A coalition, on the other hand, may be able to manipulate the result. With some rules, like STV, it is NP-hard to compute how a single agent needs to vote to manipulate the result or to prove that manipulation by this single agent is impossible (Bartholdi & Orlin, 1991). With other rules like Borda, it may require a coalition of two agents for manipulation to be NP-hard to compute (Davies, Katsirelos, Narodytska, & Walsh, 2011; Betzler, Niedermeier, & Woeginger, 2011). With other rules like veto, we may require a coalition of manipulating agents whose size is unbounded (and allowed to grow with the problem size) for manipulation to be NP-hard to compute (Conitzer et al., 2007).

Complete or incomplete information: Many results assume that the manipulator(s) have complete information about the other agents' votes. Of course, we may not know precisely how other agents will vote in practice. However, there are several reasons why the case of complete information is interesting. First, if we can show that it is computationally intractable to compute how to manipulate the election with complete information then it is also intractable when we have incomplete information. Second, the complete information case informs the case when we have uncertainty. For instance, if it is computationally intractable for a coalition to compute how to manipulate an election with complete information then it is also intractable for an individual to compute how to manipulate an election when we have only probabilistic information about the votes (Conitzer et al., 2007).

Constructive or destructive manipulation: Is the manipulator trying to make one particular candidate win (constructive manipulation) or prevent one particular candidate from winning (destructive manipulation)? Destructive manipulation is easier to compute than constructive manipulation. For instance, constructive manipulation of the veto rule by a coalition of agents with weighted votes is NP-hard but destructive manipulation is polynomial (Conitzer et al., 2007). However, there are also rules where both destructive and constructive manipulation are in the same complexity class. For example, both constructive and destructive manipulation of plurality are polynomial to compute, whilst both constructive and destructive manipulation of plurality with runoff for weighted votes are NP-hard (Conitzer et al., 2007).

In Figure 1, we give a representative selection of results about the complexity of manipulating voting. References to many of these results can be found in the work of Conitzer et al. (2007). In this paper, we will focus on two cases that cover the main types of computational complexity results that have been identified: manipulation of STV with unweighted votes by a single agent

number of candidates	unweighted votes, constructive manipulation	weighted votes, constructive 2	3	≥4	destructive 2	≥3
plurality	P	P	P	P	P	P
Borda	NP-c	P	NP-c	NP-c	P	P
veto	P	P	NP-c	NP-c	P	P
STV	NP-c	P	NP-c	NP-c	P	NP-c
plurality with runoff	P	P	NP-c	NP-c	P	NP-c
Copeland	NP-c	P	P	NP-c	P	P

Figure 1: Computational complexity of deciding if various voting rules can be manipulated by an agent (unweighted votes) or a coalition of agents (weighted votes). P means that the problem is polynomial, and NP-c that the problem is NP-complete. For example, constructive manipulation of the veto rule is polynomial for unweighted votes or for weighted votes with 2 candidates, NP-hard for 3 or more candidates. On the other hand, destructive manipulation of the veto rule is polynomial for weighted votes with a coalition of manipulating agents and 2 or more candidates.

and an unbounded number of candidates in the election, and manipulation of veto voting with weighted votes by a coalition of agents and just 3 candidates in the election. In both cases, we assume complete information about the votes of the other agents. The two cases thus cover cases in which computational complexity is associated: unweighted votes, a small (bounded) number of manipulators and a large (unbounded) number of candidates; weighted votes, a large (unbounded) number of manipulators, and a small (bounded) number of candidates.

STV is a very obvious and interesting case to consider when we study the computational complexity of manipulation. STV is one of the few voting rules used in practice where manipulation is NP-hard to compute when votes are unweighted. Bartholdi and Orlin argued that STV is one of the most promising voting rules to consider in this respect:

> *"STV is apparently unique among voting schemes in actual use today in that it is computationally resistant to manipulation."*, (Bartholdi & Orlin, 1991, p. 341).

STV also appears more difficult to manipulate than many other rules. For example, Chamberlain (1985) studied four different measures of the manipulability of a voting rule: the probability that manipulation is possible, the number of candidates who can be made to win, the coalition size necessary to manipulate, and the margin-of-error which still results in a successful manipulation. Compared to other commonly used rules like plurality and Borda, his results showed that STV was the most difficult to manipulate by a substantial margin. He concluded that:

> *"[this] superior performance ... combined with the rather complex and implausible nature of the strategies to manipulate it, suggest that it [the STV rule] may be quite resistant to manipulation"*, (Chamberlin, 1985, p. 203).

The second case considered in this paper (manipulation of the veto rule by a coalition of manipulators with three candidates) is interesting to study for several reasons. First, the veto rule is a simple representative of voting rules where manipulation by a coalition of agents with weighted votes and a small number of candidates is NP-hard to compute. Second, the veto rule is very easy to reason about. Unlike STV, there are not multiple rounds and the elimination of candidates to worry about. In fact, as we show, manipulation of the veto rule is equivalent to a simple number partitioning problem. We can therefore use efficient number partitioning algorithms to compute manipulations. Third, the veto rule is on the borderline of tractability since constructive manipulation of the rule by a coalition of weighted agents is NP-hard but destructive manipulation is polynomial (since the best way to ensure a candidate does not win is simply to veto this candidate) (Conitzer et al., 2007).

Our empirical study considers the computational difficulty of computing manipulations in practice. NP-hardness results only describe the worst-case. There is increasing concern that computational complexity results like these may not reflect the actual difficulty of computing manipulations in practice. For instance, a number of recent theoretical results suggest that manipulation may often be computationally easy (Conitzer & Sandholm, 2006; Procaccia & Rosenschein, 2007b; Xia & Conitzer, 2008a; Friedgut, Kalai, & Nisan, 2008; Xia & Conitzer, 2008b). Our results demonstrate that we can profitably study this issue empirically. There are several reasons why empirical analysis like that undertaken here is useful. First, theoretical analysis is often asymptotic so does not show the size of hidden constants. In addition, elections are typically bounded in size. Can we be sure that asymptotic behaviour is relevant for the finite sized electorates met in practice? For instance, our results suggest that we can easily compute manipulations for almost any type of STV election with up to 100 candidates. Second, theoretical analysis is often restricted to particular distributions (e.g. independent and identically distributed votes). Manipulation may be very different in practice due to correlations between votes. For instance, if all preferences are single-peaked then a voting rule which selects the median candidate is not manipulable. With the median voting rule, it is in the best interests of all agents to state their true preferences. It is thus clear that correlations between votes can have an impact on the manipulability of an election. Indeed, a number of recent results have studied whether manipulation remains computationally hard when votes are limited to be single-peaked (Walsh, 2007; Faliszewski, Hemaspaandra, Hemaspaandra, & Rothe, 2009; Brandt, Brill, Hemaspaandra, & Hemaspaandra, 2010). Our experiments will therefore look at elections in which there are correlations between votes. Third, many of the theoretical results about the computational complexity of manipulation have been hard won and are limited in their scope. For instance, a long standing open result was recently settled, proving that computing a manipulation of the Borda rule by a coalition of manipulators is NP-hard (Davies et al., 2011; Betzler et al., 2011). However, both proofs require precisely 2 manipulators. It remains open if computing a manipulation of the Borda rule by a larger coalition is NP-hard. An empirical study may quickly suggest whether the result extends to larger coalitions. Finally, empirical studies may suggest new avenues for theoretical study. For example, the experiments reported here suggest a simple and universal form for the probability that a coalition of agents in a veto election can elect a desired candidate. It would be interesting to try to derive this form theoretically.

2. Background

We give some notation and background that will be used throughout the rest of the paper. Let m be the number of *candidates* in the election. A *vote* is a linear order (a transitive, antisymmetric, and total relation) over the set of m candidates. Let n the the number of agents voting. A *profile* is a n-tuple of votes. We let $N(i, j)$ be the number of agents preferring i to j in a profile. A *voting rule* is a function that maps any profile to an unique winning alternative. In this paper, we consider a number of common voting rules:

Scoring rules: $(w_1, \ldots, w_m)$ is a vector of weights, the ith candidate in a total order scores w_i, and the winner is the candidate with highest total score. The *plurality* rule has the weight vector $(1, 0, \ldots, 0)$, the *veto* rule has the vector $(1, 1, \ldots, 1, 0)$, whilst the *Borda* rule has the vector $(m-1, m-2, \ldots, 0)$. With the veto rule, each voter effectively vetoes one candidate and the candidate with the fewest vetoes wins.

Single transferable vote (STV): STV proceeds in a number of rounds. Unless one candidate has a majority of first place votes, we eliminate the candidate with the least number of first place votes. Any ballots placing the eliminated candidate in first place are re-assigned to the second place candidate. We then repeat until one candidate has a majority.

Copeland (aka tournament): The candidate with the highest Copeland score wins. The Copeland score of candidate i is $\sum_{i \neq j}(N(i,j) > \frac{n}{2}) - (N(i,j) < \frac{n}{2})$. The Copeland winner is the candidate that wins the most pairwise elections. In the second order Copeland rule, if there is a tie, the winner is the candidate whose defeated competitors have the largest sum of Copeland scores.

Maximin (aka Simpson): The candidate with the highest maximin score wins. The maximin score of candidate i is $\min_{i \neq j} N(i,j)$. The Simpson winner is the candidate whose worst performance in pairwise elections is best.

All these rules can be easily modified to work with *weighted* votes. A vote of integer weight w can be viewed as w agents who vote identically. All these voting rules are *anonymous* as the order of votes in the profile is unimportant. A profile can therefore be thought of as a multi-set of n votes. To ensure the winner is unique, we will sometimes need to break ties (e.g. when two candidates have the same number of vetoes, or when two candidates have the same number of first place votes). In the UK, for example, when an election is tied, the returning officer will choose between the candidates using a random method like lots or a coin toss. A typical assumption made in the literature (and in this paper) is that ties are broken in favour of the manipulator. More precisely, given a choice of several candidates, we tie-break in favour of the candidate most preferred by the manipulator. Suppose the manipulator can make their preferred candidate win assuming ties are broken in their favour but ties are in fact broken at random. Then we can conclude that the manipulator can increase the chance of getting their preferred result. It would be interesting to consider other tie-breaking rules. Indeed, tie-breaking can even introduce computational complexity into manipulation. For example, computing how to manipulate the Copeland rule with weighted votes is polynomial if ties are scored with 1 but NP-hard if they are scored with 0 (Faliszewski, Hemaspaandra, & Schnoor, 2008).

We will consider one agent or a coalition of k agents trying to manipulate the result of the election. Manipulation is the situation where the manipulators vote differently to their true preferences in order to improve the outcome from their perspective. As is common in the literature, we assume that the manipulators have *complete* knowledge of the other votes (and, where appropriate, complete knowledge of all the weights associated with all the votes). Whilst it may be unrealistic in practice to assume we have complete knowledge about the other votes, there are several reasons why this case is interesting to consider. First, complete information is likely to be a special case of any uncertainty model. Hence, any computational hardness results for complete information directly imply hardness for the corresponding uncertainty model. Second, results about the hardness of manipulation by a coalition with weighted votes and complete information imply hardness of manipulation by an individual agent with unweighted votes and incomplete information (Conitzer et al., 2007). Third, by assuming complete information, we factor out any complexity coming from the uncertainty model and focus instead on computing just the manipulation.

In addition to some of the standard uniform and random models of votes, we will consider two restricted types of votes: single-peaked and single-troughed votes. With single-peaked votes, candidates can be placed on a line, and an agent's preference for a candidate decreases with distance from their single most preferred candidate. Single-peaked preferences are interesting from several perspectives. First, single-peaked preferences are likely to occur in a number of domains. For example, if you are buying a house, you might have an optimal price in mind and your preference for a house decreases as the distance from this price increases. Second, single-peaked preferences are easy to elicit. Conitzer (2007, 2009) gives a strategy for eliciting any single-peaked preference ordering with a linear number of pairwise ranking questions. Third, single-peaked preferences prevent certain problematic situations arising when aggregating preferences. In particular, they prevent the existence of Condorcet cycles. In fact, the *median* candidate of a single-peaked profile beats all others in pairwise comparisons (that is, the median candidate is the Condorcet winner) (Black, 1948). With single-troughed votes, on the other hand, candidates can be placed on a line, and an agent's preference for a candidate increases with distance from their single least preferred candidate. For example, if the candidates are locations to build a new incinerator, you might have a least preferred location (your own neighbourhood), and your preference increases the further away the incinerator is from this. Single-troughed votes have similar nice properties to single-peaked votes (Barberà, Berga, & Moreno, 2009).

3. Single Transferable Voting

We begin with an empirical study of manipulation in STV elections. STV is used in a wide variety of real-world settings including the election of the Irish presidency, the Australian House of Representatives, the Academy awards, and many organisations including the American Political Science Association, the International Olympic Committee, and the British Labour Party. Interestingly it is NP-hard to compute if a single agent can manipulate the STV rule. Indeed, it is one of the few voting rules used in practice where manipulation is NP-hard to compute in this setting. More precisely, STV is NP-hard to manipulate by a single agent if the number of candidates is unbounded and votes are unweighted (Bartholdi & Orlin, 1991), or by a coalition of agents if there are 3 or more candidates and votes are weighted (Conitzer et al., 2007). Coleman and Teague give an enumerative method for a coalition of k unweighted agents to compute a manipulation of the STV rule which runs in $O(m!(n+mk))$ time where n is the number of agents voting and m is the number of candidates (Coleman & Teague, 2007). For a single manipulator, Conitzer, Sandholm and Lang give an $O(n1.62^m)$ time algorithm (called CSL from now on) to compute the set of candidates that can win an STV election (Conitzer et al., 2007).

In Figure 2, we give a modified version of the CSL algorithm for computing a manipulation of the STV rule. This uses a similar recursion as CSL but changes the original algorithm in several ways to take advantage of the fact that we only want to compute if one distinguished candidate can win and do not need to know all of the other candidates that can possibly win. There are two main changes to CSL. First, we can ignore elections in which the chosen candidate is eliminated. Second, we can terminate search as soon as a manipulation is found in which the chosen candidate wins. In particular, we do not need to explore the left branch of the search tree when the right branch gives a successful manipulation.

We tested the modified algorithm with the simplest possible scenario: elections in which each vote is equally likely. We have one agent trying to manipulate an election of m candidates where n other agents vote. Votes are drawn uniformly at random from all $m!$ possible votes. This is the Impartial Culture (IC) model. To show the benefits of our modifications to the CSL algorithm, we ran a simple experiment in which $m = n$. The experiment was run in CLISP 2.42 on a 3.2 GHz Pentium 4 with 3GB of memory running Ubuntu 8.04.3. Table 1 gives the mean nodes explored and runtime needed to compute a manipulation or prove none exists. Median and other percentiles display similar behaviour. We see that our modified method can be considerably faster than the original CSL algorithm. In addition, as problems get larger, the improvement increases. At $n = 32$, our method is nearly 10 times faster than CSL. This increases to roughly 40 times faster at $n = 128$. These improvements reduce the time to find a manipulation on the largest problems from several hours to a couple of minutes.

3.1 Varying Number of Agents

We next performed some detailed experiments looking for phase transition behaviour and hard manipulation problems. In many other NP-hard problem domains, the computationally hard instances have been shown to be often associated with the region of the parameter space in which there is a rapid transition in the probability that a solution exists (Cheeseman, Kanefsky, & Taylor, 1991;

```
MANIPULATE(c, R, (s_1, ..., s_m), f)
if |R| = 1                                   ; Is there one candidate left?
   then return (R = {c})                     ; Is it the chosen candidate?
if f = 0                                     ; Is the top of the manipulator's vote currently free?
   then
        d ← arg min_{j∈R}(s_j)               ; Who will currently be eliminated?
        s_d ← s_d + w                        ; Suppose the manipulator votes for them
        e ← arg min_{j∈R}(s_j)
        if d = e                             ; Does this not change the result?
           then return
                (c ≠ d) and MANIPULATE(c, R − {d}, Transfer((s_1, ..., s_m), d, R), 0)
           else  return
                ((c ≠ d) and MANIPULATE(c, R − {d}, Transfer((s_1, ..., s_m), d, R), 0)) or
                ((c ≠ e) and MANIPULATE(c, R − {e}, Transfer((s_1, ..., s_m), e, R), d))
   else                                      ; The top of the manipulator's vote is fixed
        d ← arg min_{j∈R}(s_j)               ; Who will be eliminated?
        if c = d                             ; Is this the chosen candidate?
           then return false
        if d = f                             ; Is the manipulator free again to change the result?
           then return MANIPULATE(c, R − {d}, Transfer((s_1, ..., s_m), d, R), 0)
           else  return MANIPULATE(c, R − {d}, Transfer((s_1, ..., s_m), d, R), f)
```

Figure 2: Our modified algorithm to compute if an agent can manipulate an STV election. We use integers from 1 to m for the candidates, integers from 1 to n for the agents (with n being the manipulator), c for the candidate who the manipulator wants to win, R for the set of un-eliminated candidates, s_j for the weight of agents ranking candidate j first amongst R, w for the weight of the manipulator, and f for the candidate most highly ranked by the manipulator amongst R (or 0 if there is currently no constraint on who is most highly ranked). The function *Transfer* computes the a vector of the new weights of agents ranking candidate j first amongst R after a given candidate is eliminated. The algorithm is initially called with R set to every candidate, and f to 0.

Mitchell, Selman, & Levesque, 1992). The "phase transition" between a satisfiable and an unsatisfiable phase resembles that seen in statistical physics in Ising magnets and similar systems. There are several good surveys of this area (Dubois, Monasson, Selman, & Zecchina, 2001; Hartmann & Weigt, 2005; Gomes & Walsh, 2006).

Our first experiment varied the number of agents voting. In Figures 3 and 4, we plot the probability that a manipulator can make a random agent win, and the cost to compute if this is possible when we fix the number of candidates but vary the number of agents in the STV election. In this and subsequent experiments, we tested 1000 problems at each point. The number of candidates and of agents voting are varied in powers of 2, typically from 2^0 $(= 1)$ to 2^7 $(= 128)$ unless otherwise indicated.

The ability of an agent to manipulate the election decreases as the number of agents increases. Only if there are few votes and few candidates is there a significant chance that the manipulator will be able to change the result. Phase transition behaviour has been observed in many NP-complete problem domains including propositional satisfiability (Mitchell et al., 1992; Gent & Walsh, 1994, 1996b), constraint satisfaction (Prosser, 1994; Smith, 1994; Gent, MacIntyre, Prosser, & Walsh, 1995; Gent, MacIntyre, Prosser, Smith, & Walsh, 2001), graph colouring (Walsh, 1998, 1999, 2001), number partitioning (Gent & Walsh, 1996a, 1998; Mertens, 2001) and the travelling salesperson problem (Zhang & Korf, 1996; Gent & Walsh, 1996c). Unlike these domains, the probability curve observed

n	CSL algorithm nodes	time/s	Modified algorithm nodes	time/s
2	1.46	**0.00**	**1.24**	**0.00**
4	3.28	**0.00**	**1.59**	**0.00**
8	11.80	**0.00**	**3.70**	**0.00**
16	59.05	0.03	**12.62**	**0.01**
32	570.11	0.63	**55.20**	**0.09**
64	14,676.17	33.22	**963.39**	**3.00**
128	8,429,800.00	6,538.13	**159,221.10**	**176.68**

Table 1: Comparison between the original CSL algorithm and the modified version which computes a constructive manipulation of an STV election. The table gives the mean nodes explored and runtime needed to compute a manipulation or prove none exists. Median and other percentiles display similar behaviour. Each election has n agents voting uniformly at random over n different candidates. Best results in each row are in **bold**.

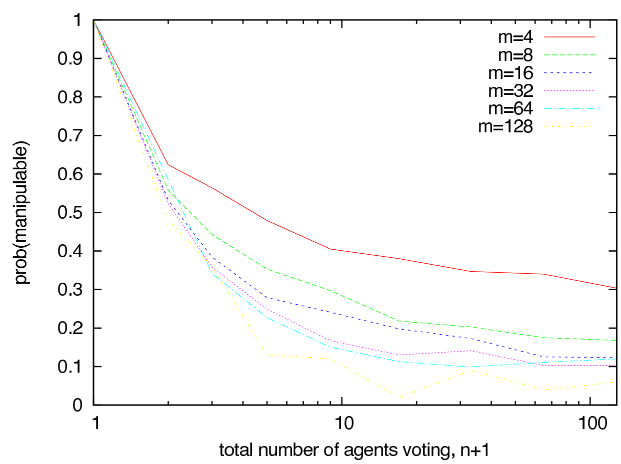

Figure 3: Manipulability of an STV election containing random uniform votes. The number of candidates is fixed and we vary the number of agents voting. The vertical axis measures the probability that a single manipulating agent can make a random candidate win. The horizontal axis measures the total number of agents voting. Note that n is the number of agents voting besides the manipulator so that a log scale can be used on the horizontal axis.

here does not appear to sharpen to a step function around a fixed point. The probability curve resembles the smooth phase transitions seen in polynomial problems like 2-colouring (Achlioptas, 1999) and 1-in-2 satisfiability (Walsh, 2002). As indicated before, we assume that ties are broken in favour of the manipulator. For this reason, the probability that an election is manipulable is greater than $\frac{1}{m}$. Finding a manipulation or proving none is possible is easy unless we have both a large

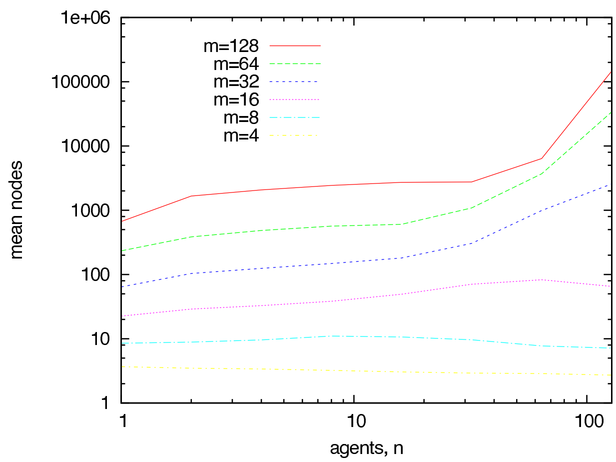

Figure 4: Search cost to compute if an agent can manipulate an STV election containing random uniform votes. The number of candidates, m is fixed and we vary the number of agents. The vertical axis measures the mean nodes explored to compute if a single manipulating agent can make a random candidate win. The horizontal axis measures the number of agents voting besides the manipulator. Median and other percentiles are similar.

number of agents and a large number of candidates. However, in this situation, the chance that the manipulator can change the result is very small.

3.2 Varying Number of Candidates

Our next experiment slices the parameter space in an orthogonal direction, varying the number of candidates in the election. In Figure 5, we plot the cost to compute if the manipulator can make a random agent win an STV election when we fix the number of agents but vary the number of candidates. The probability curve that the manipulator can make a random agent win resembles Figure 3. Whilst the cost of computing a manipulation appears to increase exponentially with the number of candidates m, the observed scaling is much better than the 1.62^m worst case scaling of the original CSL algorithm. We can easily compute manipulations for up to 128 candidates. Note that 1.62^m is over 10^{26} for $m = 128$. Thus, we appear to be far from the worst case. We fitted the observed data to the function cd^m and found a fit with $d = 1.008$ and a coefficient of determination $R^2 = 0.95$ indicating a good fit.

3.3 Correlated Votes

In many real life situations, votes are not completely uniform and uncorrelated with each other. What happens if we introduce correlation between votes? Here we consider random votes drawn from the Polya-Eggenberger urn model (Berg, 1985). In this model, we have an urn containing all $m!$ possible votes. We draw votes out of the urn at random, and put them back into the urn with

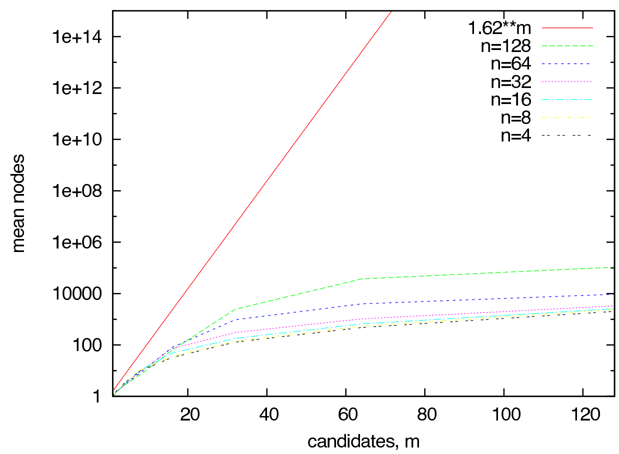

Figure 5: Search cost to compute if an agent can manipulate an STV election containing random uniform votes. The number of agents, n is fixed and we vary the number of candidates. The vertical axis measures the mean number of nodes explored to compute a manipulation or prove that none exists. The horizontal axis measures m, the number of candidates in the election. Median and other percentiles are similar.

a additional votes of the same type (where a is a parameter). As a increases, there is increasing correlation between the votes. This generalises both the Impartial Culture model ($a = 0$) in which all votes are equally likely and the Impartial Anonymous Culture ($a = 1$) model in which all profiles are equally likely (McCabe-Dansted & Slinko, 2006). To give a parameter independent of problem size, we consider $b = \frac{a}{m!}$. For instance, with $b = 1$, there is a 50% chance that the second vote is the same as the first.

In Figures 6 and 7, we plot the probability that a manipulator can make a random agent win an STV election, and the cost to compute if this is possible as we vary the number of candidates in an election where votes are drawn from the Polya-Eggenberger urn model. As before, the ability of an agent to manipulate the election decreases as the number of candidates, m increases. The search cost to compute a manipulation appears to increase exponentially with the number of candidates m. However, we can easily compute manipulations for up to 128 candidates. We fitted the observed data to the function cd^m and found a fit with $d = 1.001$ and a coefficient of determination $R^2 = 0.99$ indicating a good fit.

In Figure 8, we plot the cost to compute a manipulation when we fix the number of candidates but vary the number of agents. As before, the ability of an agent to manipulate the election decreases as the number of agents increases. Only if there are few votes and few candidates is there any chance that the manipulator will succeed. As in previous experiments, finding a manipulation or proving none exists is easy even if we have many agents and candidates. We also observed results which are almost indistinguishable when votes were correlated by being single-peaked (or single-troughed) and were drawn either uniformly at random or from an urn model.

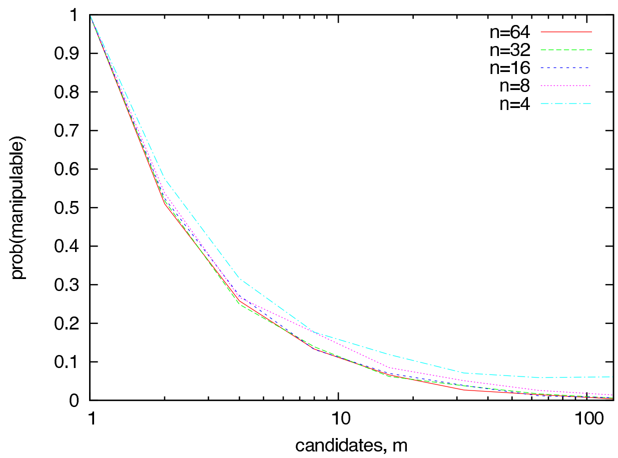

Figure 6: Manipulability of an STV election containing correlated votes. The number of agents is fixed and we vary the number of candidates, m. The n fixed votes are drawn from the Polya-Eggenberger urn model with $b = 1$. The vertical axis measures the probability that the manipulator can make a random candidate win. The horizontal axis measures the number of candidates, m in the election.

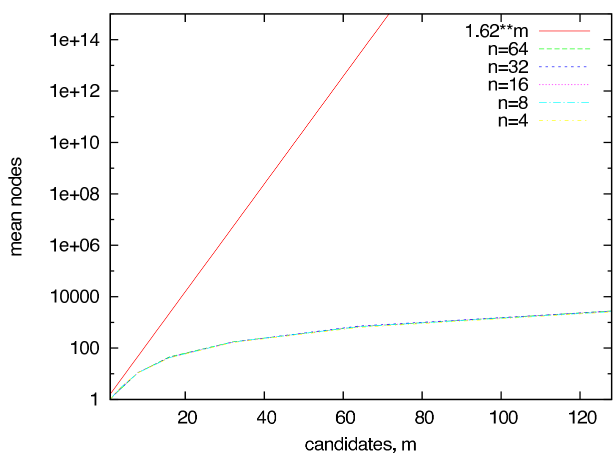

Figure 7: Search cost to compute if an agent can manipulate an STV election containing correlated votes. The number of agents, n is fixed and we vary the number of candidates, m. The n fixed votes are drawn using the Polya-Eggenberger urn model with $b = 1$. The vertical axis measures the mean number of search nodes explored to compute a manipulation or prove that none exists. The horizontal axis measures the number of candidates, m in the election. The curves for different n fit closely on top of each other. Median and other percentiles are similar.

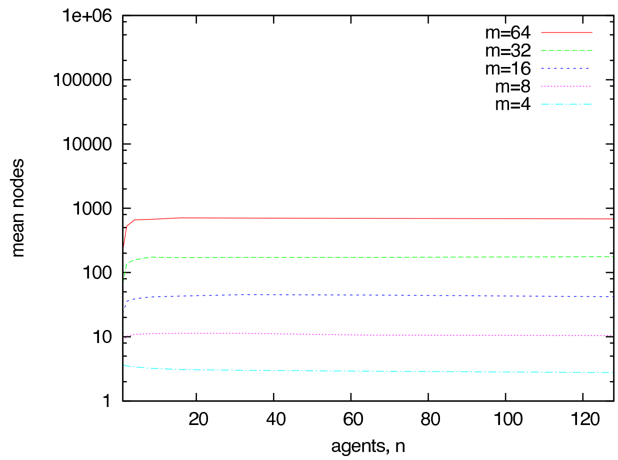

Figure 8: Search cost to compute if an agent can manipulate an STV election with correlated votes. The number of candidates, m is fixed and we vary the number of agents, n. The n fixed votes are drawn using the Polya-Eggenberger urn model with $b = 1$. The vertical axis measures the mean number of search nodes explored to compute a manipulation or prove that none exists. The horizontal axis measures the number of agents, n. Median and other percentiles are similar.

4. Coalition Manipulation

Our algorithm for computing manipulation of an STV election by a single agent can also be used to compute if a coalition can manipulate an STV election when the members of coalition vote in unison. This ignores more complex manipulations where the members of the coalition need to vote in different ways. Insisting that the members of the coalition vote in unison might be reasonable if we wish manipulation to have both a low computational and communication cost (Slinko & White, 2008). In Figures 9 and 10, we plot the probability that a coalition voting in unison can make a random agent win an STV election, and the cost to compute if this is possible as we vary the size of the coalition. Theoretical results due to Procaccia and Rosenschein (2007a) and Xia and Conitzer (2008a) suggest that the critical size of a coalition that can manipulate an election grows as $\sqrt{n}$. We therefore normalize the coalition size by $\sqrt{n}$.

The ability of the coalition to manipulate the election increases as the size of the coalition increases. When the coalition is about $\sqrt{n}$ in size, the probability that the coalition can manipulate the election so that a candidate chosen at random wins is around $\frac{1}{2}$. The cost to compute a manipulation (or prove that none exists) decreases as we increase the size of the coalition. It is easier for a larger coalition to manipulate an election than for a smaller one.

These experiments again suggest different behaviour occurs here than in other combinatorial problems like propositional satisfiability and graph colouring. For instance, we do not see a rapid transition that sharpens around a fixed point as in 3-satisfiability. When we vary the coalition size, we see a transition in the probability of being able to manipulate the result around a coalition size $k = \sqrt{n}$. However, this transition appears smooth and does not seem to sharpen towards a step function as n increases. In addition, hard instances do not occur around $k = \sqrt{n}$. Indeed, the hardest instances are when the coalition is smaller than this and has only a small chance of being able to manipulate the result.

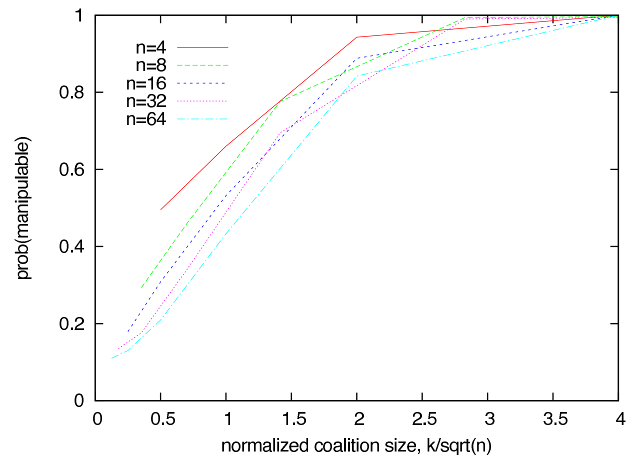

Figure 9: Manipulability of an STV election as we vary the size of the manipulating coalition. The number of candidates is the same as the number of non-manipulating agents. The n fixed votes are uniformly drawn at random from the $n!$ possible votes. The vertical axis measures the probability that the coalition can make a random candidate win. The horizontal axis measures the coalition size, k normalized by $\sqrt{n}$.

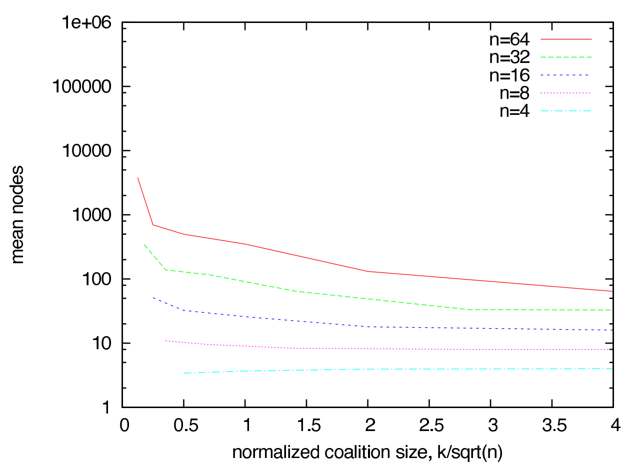

Figure 10: Search cost to compute if a coalition can manipulate an STV election as we vary coalition size. The vertical axis measures the mean number of search nodes explored to compute a manipulation or prove that none exists. The horizontal axis measures the coalition size, k normalized by $\sqrt{n}$. Median and other percentiles are similar.

5. Sampling Real Elections

Elections met in practice may differ from those sampled so far. There might, for instance, be some votes which are never cast. On the other hand, with the models studied so far every possible vote has a non-zero probability of being seen. We therefore sampled some real voting records. We have previously studied phase transition behaviour in other real world problems using similar sampling techniques (Gent & Walsh, 1995; Gent, Hoos, Prosser, & Walsh, 1999).

Our first experiment uses the votes cast by 10 teams of scientists to select one of 32 different trajectories for NASA's Mariner spacecraft (Dyer & Miles, 1976). Each team ranked the different trajectories based on their scientific value. We sampled these votes. For elections with 10 or fewer agents voting, we simply took a random subset of the 10 votes. For elections with more than 10 agents voting, we simply sampled from the 10 votes with uniform frequency. For elections with 32 or fewer candidates, we simply took a random subset of the 32 candidates. Finally for elections with more than 32 candidates, we duplicated each candidate and assigned them the same ranking. Since STV works on total orders, we then forced each agent to break any ties randomly. Each agent broke ties independently of any other agent. New candidates introduced in this way are *clones* of the existing candidates. It would be interesting to consider other, perhaps more random methods for introducing new candidates. Nevertheless, we note that "clones" are a feature of a number of real world elections. Indeed, one way to manipulate an election is to introduce clone candidates for the opposition, and thereby to divide their vote. For example, as motivation for studying clones, Tideman (1987) describes how he successfully won the vote for class treasurer as a somewhat precocious 12 year old by nominating the best friend of his main rival. We therefore believe it may be of interest to consider elections like those generated here in which clones can be present.

In Figures 11 to 12, we plot the cost to compute if a manipulator can make a random agent win an STV election as we vary the number of candidates and agents. Votes are sampled from the NASA experiment as explained earlier. The probability that the manipulator can manipulate the election resembles the probability curve for uniform random votes. The search needed to compute a manipulation again appears to increase exponentially with the number of candidates m. However, the observed scaling is much better than the 1.62^m worst-case scaling of the original CSL algorithm. We can easily compute manipulations for up to 128 candidates.

In our second experiment, we used votes from a faculty hiring committee at the University of California at Irvine (Dobra, 1983). This dataset had 10 votes for 3 different candidates. We sampled from this data set in the same ways as from the NASA dataset and observed very similar results. Results are reported in Figures 13 and 14. As in the previous experiments, it was easy to find a manipulation or prove that none exists. The observed scaling was again much better than the 1.62^m worst-case scaling of the original CSL algorithm.

6. Veto Rule

We now turn to the manipulation of elections where there is a small, bounded number of candidates, the votes are weighted and there is a coalition of agents trying to manipulate the result. For this part of the empirical study, we consider the veto rule. We recall that veto is a scoring rule in which each agent gets to cast a veto against one candidate. The candidate with the fewest vetoes wins. As the next theorem shows, simple number partitioning algorithms can be used to compute a successful manipulation of the veto rule. More precisely, as the following theorem demonstrates, the manipulation of an election with 3 candidates and weighted votes by a coalition (which is NP-hard to compute) can be directly reduced to 2-way number partitioning problem. We therefore compute manipulations in our experiments using an efficient number partitioning algorithm like that proposed by Korf (1995).

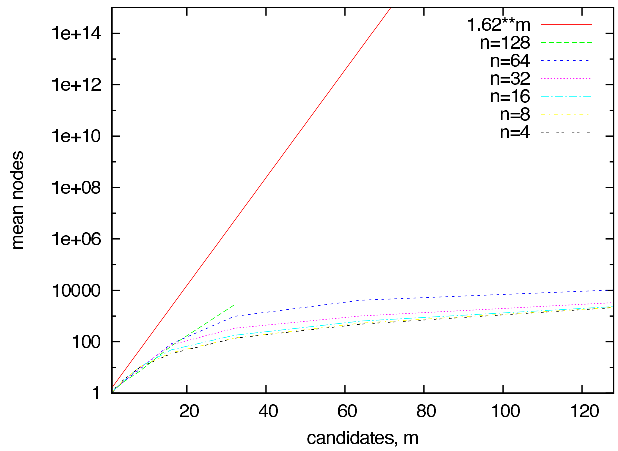

Figure 11: Search cost to compute if an agent can manipulate an STV election with votes sampled from the NASA experiment. The number of agents, n is fixed and we vary the number of candidates, m. The vertical axis measures the mean number of search nodes explored to compute a manipulation or prove that none exists. The horizontal axis measures the number of candidates, m. Median and other percentiles are similar.

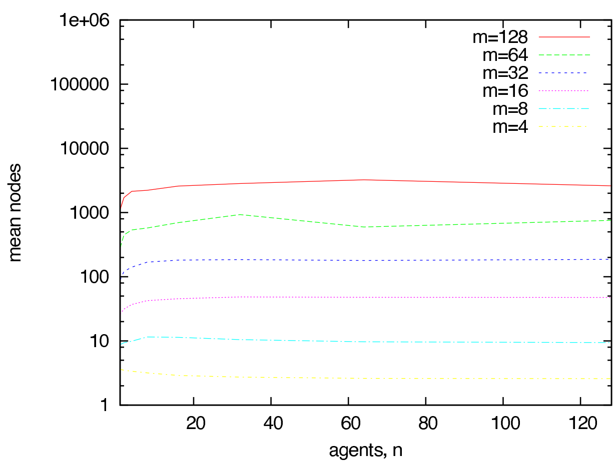

Figure 12: Search cost to compute if an agent can manipulate an STV election with votes sampled from the NASA experiment. The number of candidates, m is fixed and we vary the number of agents, n. The vertical axis measures the mean number of search nodes explored to compute a manipulation or prove that none exists. The horizontal axis measures the number of agents, n. Median and other percentiles are similar.

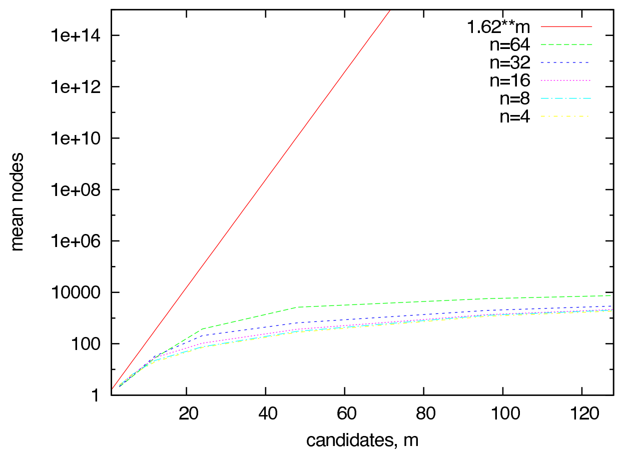

Figure 13: Search cost to compute if an agent can manipulate an STV election with votes sampled from a faculty hiring committee. The number of agents voting, n is fixed and we vary the number of candidates, m. The vertical axis measures the mean number of search nodes explored to compute a manipulation or prove that none exists. The horizontal axis measures the number of candidates, m. Median and other percentiles are similar.

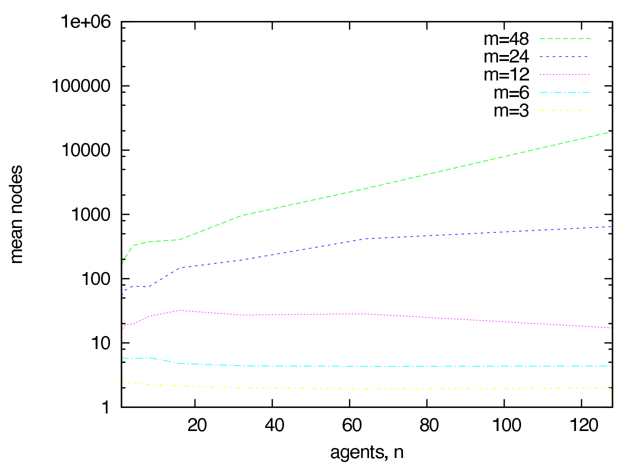

Figure 14: Search cost to compute if an agent can manipulate an STV election with votes sampled from a faculty hiring committee. The number of candidates, m is fixed and we vary the number of agents voting, n. The vertical axis measures the mean number of search nodes explored to compute a manipulation or prove that none exists. The horizontal axis measures the number of agents, n. Median and other percentiles are similar.

Theorem 1 *There exists a successful manipulation of an election with 3 candidates by a weighted coalition using the veto rule if and only if there exists a partitioning of* $W \cup \{|a-b|\}$ *into two bags such that the difference between their two sums is less than or equal to* $a+b-2c+\sum_{i \in W} i$*, where* W *is the multiset of weights of the manipulating coalition,* a*,* b *and* c *are the weights of vetoes assigned to the three candidates by the non-manipulators and the manipulators wish the candidate with weight* c *to win.*

Proof: It never helps a coalition manipulating the veto rule to veto the candidate that they wish to win. The coalition does, however, need to decide how to divide their vetoes between the candidates that they wish to lose. Consider the case $a \geq b$. Suppose the partition has weights $w - \Delta/2$ and $w + \Delta/2$ where $2w = \sum_{i \in W \cup \{|a-b|\}} i$ and Δ is the difference between the two sums. The same partition of vetoes is a successful manipulation if and only if the winning candidate has no more vetoes than the next best candidate. That is, $c \leq b + (w - \Delta/2)$. Hence $\Delta \leq 2w + 2b - 2c = (a-b) + 2b - 2c + \sum_{i \in W} i = (a+b-2c) + 2\sum_{i \in W} i$. In the other case, $a < b$ and $\Delta \leq (b+a-2c) + \sum_{i \in W} i$. Thus $\Delta \leq a+b-2c+\sum_{i \in W} i$. $\Box$

As with the STV rule, we start our analysis with uniform votes. We first consider the case that the n agents veto uniformly at random one of the 3 possible candidates, and vetoes carry weights drawn uniformly from $(0, w]$. When the coalition is small in size, it has too little weight to be able to change the result. On the other hand, when the coalition is large in size, it is sure to be able to make a favoured candidate win. There is thus a transition in the manipulability of the problem as the coalition size increases (see Figure 15).

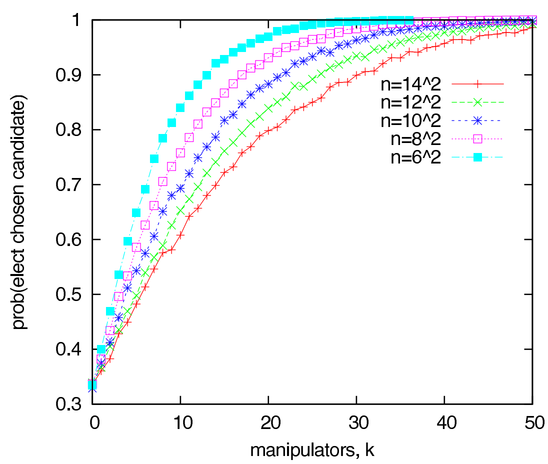

Figure 15: Manipulability of a veto election. The vertical axis measures the probability that a coalition of k agents elect a chosen candidate in a veto election where n agents have already voted. The horizontal axis measures the number of manipulators, k. Vetoes are weighted and weights are uniformly drawn from $(0, 2^8]$. At $k = 0$, there is a 1/3rd chance that the non-manipulators have already elected this candidate.

Based on the work of Procaccia and Rosenschien (2007a) and of Xia and Conitzer (2008a), we expect the critical coalition size to increase as $\sqrt{n}$. In Figure 16, we see that the phase transition displays a simple and universal form when plotted against $k/\sqrt{n}$. The phase transition appears to be smooth, with the probability varying slowly and not approaching a step function as problem size increases. We obtained a good fit with $1 - \frac{2}{3}e^{-k/\sqrt{n}}$. Other smooth phase transitions have been seen with 2-colouring (Achlioptas, 1999), 1-in-2 satisfiability and Not-All-Equal 2-satisfiability

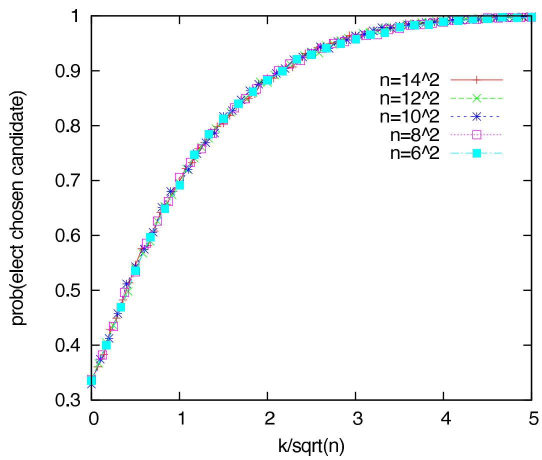

Figure 16: Manipulability of a veto election with rescaled axes. The vertical axis measures the probability that a coalition of k agents can elected a chosen candidate in a veto election where n agents have already voted. The horizontal axis measures the number of manipulators, k divided by the square root of the number of agents who have already voted. Vetoes are weighted and weights are uniformly drawn from $(0, 2^8]$. Note that the horizontal axis is scaled by $1/\sqrt{n}$ compared to the previous figure.

(Achlioptas, Chtcherba, Istrate, & Moore, 2001; Walsh, 2002). It is interesting to note that all these decision problems are polynomial.

The theoretical results mentioned earlier leave open how hard it is to compute whether a manipulation is possible when the coalition size is critical. Figure 17 displays the computational cost to find a manipulation (or prove none exists) using Korf's efficient number partitioning algorithm. Even in the critical region where problems may or may not be manipulable, it is easy to compute whether the problem is manipulable. All problems are solved in a few branches. This contrasts with phase transition behaviour in NP-complete problems like propositional satisfiability or in other complexity classes (Gent & Walsh, 1999; Bailey, Dalmau, & Kolaitis, 2001; Slaney & Walsh, 2002) where the hardest problems tend to occur around the phase transition.

7. Why Hard Veto Problems Are Rare

Based on our reduction of manipulation problems to number partitioning, we give a heuristic argument why hard manipulation problems become vanishing rare as $n \rightsquigarrow \infty$ and $k = \Theta(\sqrt{n})$. The basic idea is simple: by the time the coalition is large enough to be able to change the result, the variance in scores between the candidates is likely to be so large that computing a successful manipulation or proving none is possible will be easy. Our argument is approximate. For example, we replace discrete sums with continuous integrals, and call upon limiting results like the Central Limit Theorem. Nevertheless, it provides insight into why manipulations are typically easy to compute.

Suppose that n vetoes voted by the non-manipulators carry weights drawn uniformly from $[0, w]$. Suppose also that the k manipulators also have weights drawn uniformly from $[0, w]$, that they want candidates A and B to lose so that C wins, and that they have cast vetoes of weight a, b and c for A, B and C respectively. Without loss of generality we suppose that $a \geq b$. There are three cases to consider. In the first case, $a \geq c$ and $b \geq c$. It is then easy for the manipulators to make

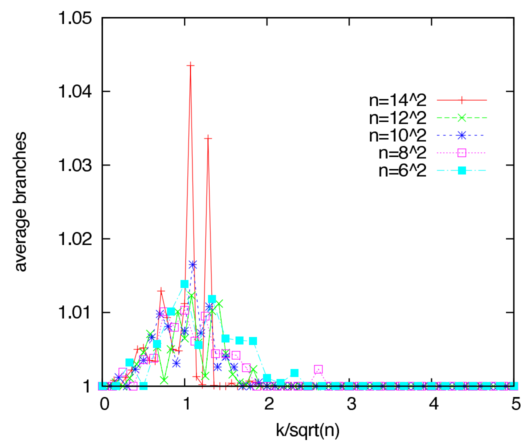

Figure 17: Computational cost for Korf's number partitioning algorithm to decide if a coalition of k agents can manipulate a veto election where n agents have already voted. Vetoes are weighted and weights are uniformly drawn from $(0, 2^k]$. The vertical axis measures the mean number of branches used by the algorithm to find a manipulation or prove none exists. As in the previous figure, the horizontal axis measures the number of manipulators, k divided by the square root of the number of agents who have already voted. We note that all problems are solved with little search. Most took a single branch to solve. Only a few took 2 or more branches.

C win since C wins whether they veto A or B. In the second case, $a \geq c > b$. Again, it is easy for the manipulators to decide if they can make C win. They all veto B. There is a successful manipulation if and only if C now wins. In the third case, $a < c$ and $b < c$. The manipulators must partition their k vetoes between A and B so that the total vetoes received by A and B exceeds those for C. Let d be the deficit in weight between A and C and between B and C. That is, $d = (c - a) + (c - b) = 2c - a - b$. We can approximate d by the sum of n random variables drawn uniformly with probability $1/3$ from $[0, 2w]$ and with probability $2/3$ from $[-w, 0]$. These variables have mean 0 and variance $2w^2/3$. By the Central Limit Theorem, d tends to a normal distribution with mean 0, and variance $t^2 = 2nw^2/3$. For a manipulation to be possible, d must be less than s, the sum of the weights of the vetoes of the manipulators. By the Central Limit Theorem, s also tends to a normal distribution with mean $\mu = kw/2$, and variance $\sigma^2 = 2kw^2/3$.

A simple heuristic argument due to (Karmarkar, Karp, Lueker, & Odlyzko, 1986) and also based on the Central Limit Theorem upper bounds the optimal partition difference of k numbers from $[0, w]$ by $O(w\sqrt{k}/2^k)$. In addition, based on the phase transition in number partitioning (Gent & Walsh, 1998), we expect partitioning problems to be easy unless $\log_2(w) = \Theta(k)$. Combining these two observations, we expect hard manipulation problems when $0 \leq s - d \leq \alpha\sqrt{k}$ for some constant α. The probability of this occurring is:

$$\int_0^\infty \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \int_{x-\alpha\sqrt{k}}^{x} \frac{1}{\sqrt{2\pi}t} e^{-\frac{y^2}{2t^2}} \, dy \, dx$$

By substituting for t, μ and σ, we get:

$$\int_0^\infty \frac{1}{\sqrt{4\pi k w^2/3}} e^{-\frac{(x-kw/2)^2}{4kw^2/3}} \int_{x-\alpha\sqrt{k}}^{x} \frac{1}{\sqrt{4\pi n w^2/3}} e^{-\frac{y^2}{4nw^2/3}} \, dy \, dx$$

For $n \rightsquigarrow \infty$, this tends to:

$$\int_0^\infty \frac{1}{\sqrt{4\pi k w^2/3}} e^{-\frac{(x-kw/2)^2}{4kw^2/3}} \frac{\alpha\sqrt{k}}{\sqrt{4\pi n w^2/3}} e^{-\frac{x^2}{4nw^2/3}} \; dx$$

As $e^{-z} \leq 1$ for $z > 0$, this is upper bounded by:

$$\frac{\alpha\sqrt{k}}{\sqrt{4\pi n w^2/3}} \int_0^\infty \frac{1}{\sqrt{4\pi k w^2/3}} e^{-\frac{(x-kw/2)^2}{4kw^2/3}} \; dx$$

Since the integral is bounded by 1, $k = \Theta(\sqrt{n})$ and $\log_2(w) = \Theta(k)$, this upper bound varies as:

$$O(\frac{1}{\sqrt{k}2^k})$$

Thus, we expect hard instances of manipulation problems to be exponentially rare. Since even a brute force manipulation algorithm takes $O(2^k)$ time in the worst-case, we do not expect the hard instances to have a significant impact on the average-case as n (and thus k) grows. We stress this is only a heuristic argument. It makes many assumptions about the complexity of manipulation problems (in particular that hard instances should lie within the narrow interval $0 \leq s - d \leq \alpha\sqrt{k}$). These assumptions are currently only supported by empirical observation and informal argument. However, the experimental results reported in Figure 17 support these conclusions.

8. Other Distributions of Vetoes

The theoretical analyses of manipulation due to Procaccia and Rosenschein (2007a) and Xia and Conitzer (2008a) suggest that the probability of an election being manipulable is largely independent of w, the size of the weights attached to the vetoes. Figure 18 demonstrates that this indeed appears to be the case in practice. When weights are varied in size from 2^8 to 2^{16}, the probability does not appear to change. In fact, the probability curve fits the same simple and universal form plotted in Figure 16. We also observed that the cost of computing a manipulation or proving that none is possible did not change as the weights were varied in size.

Similarly, theoretical results typically place few assumptions about the distribution of votes. For example, the results of Procaccia and Rosenschein (2007a) and Xia and Conitzer (2008a) that there is a critical coalition size that increases as $\Theta(\sqrt{n})$ hold for any independent and identically distributed random votes. Similarly, our heuristic argument about why hard manipulation problems are vanishingly rare depends on application of the Central Limit Theorem. It therefore works with other types of independent and identically distributed random votes.

We considered therefore another type of independent and identically distributed vote. In particular, we study an election in which weights are independently drawn from a normal distribution. Figure 19 shows that there is again a smooth phase transition in manipulability. We also plotted Figure 19 on top of Figures 16 and 18. All curves appear to fit the same simple and universal form. As with uniform weights, the computational cost of deciding if an election is manipulable was small even when the coalition size was critical. Finally, we varied the parameters of the normal distribution. The probability of electing a chosen candidate as well as the cost of computing a manipulation did not appear to depend on the mean or variance of the distribution. We do not reproduce the figures as they look identical to the previous figures.

9. Correlated Vetoes

We conjecture that one place to find hard manipulation problems for veto voting is when votes are highly correlated. For example, consider a "hung" election where all n agents veto the candidate

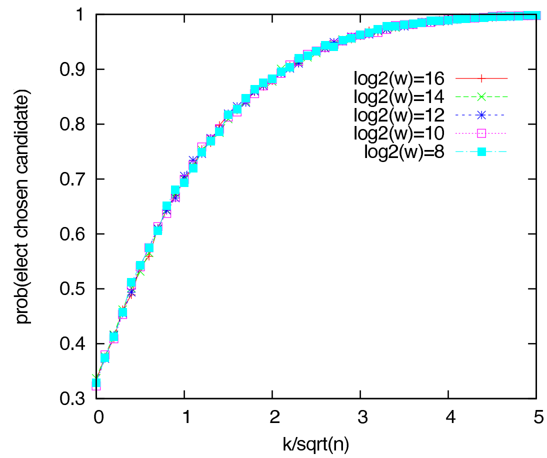

Figure 18: Independence of the size of the weights and the manipulability of a veto election. The vertical axis measures the probability that a coalition of k agents can elect a chosen candidate where n agents have already voted. As in the previous figure, the horizontal axis measures the number of manipulators, k divided by the square root of the number of agents who have already voted. Vetoes are weighted and weights are uniformly drawn from $(0, w]$.

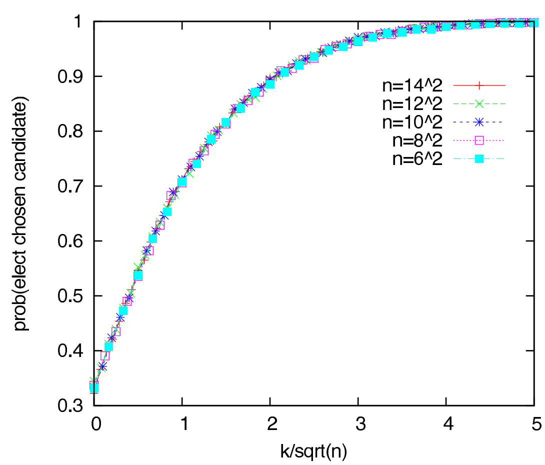

Figure 19: Manipulability of a veto election with weighted votes taken from a normal distribution. The vertical axis measure the probability that a coalition of k agents can elect a chosen candidate in a veto election where n agents have already voted. As in the previous figure, the horizontal axis measures the number of manipulators, k divided by the square root of the number of agents who have already voted. Vetoes are weighted and drawn from a normal distribution with mean 2^8 and standard deviation 2^7.

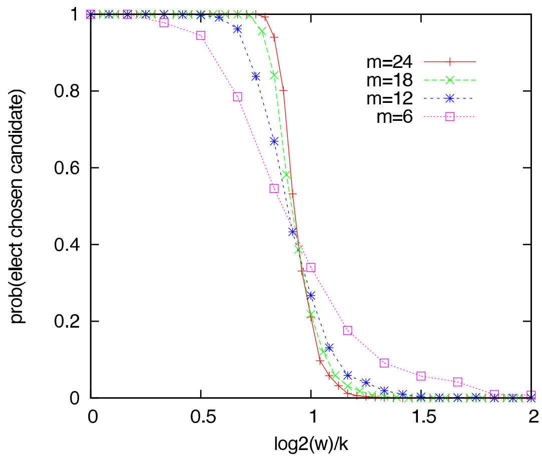

Figure 20: Manipulability of a veto election where votes are highly correlated and the result is "hung". Vetoes of the manipulators are weighted and weights are uniformly drawn from $(0, w]$, the other agents have all vetoed the candidate that the manipulators wish to win, and the sum of the weights of the manipulators is twice that of the non-manipulators. The vertical axis measures the probability that a coalition of k agents can elect a chosen candidate. The horizontal axis measures $\log_2(w)/k$.

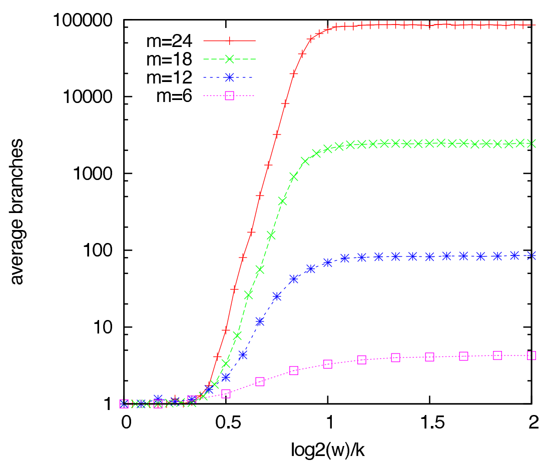

Figure 21: The search cost to decide if a "hung" veto election can be manipulated. Vetoes of the manipulators are weighted and weights are uniformly drawn from $(0, w]$, the other agents have all vetoed the candidate that the manipulators wish to win, and the sum of the weights of the manipulators is twice that of the non-manipulators. The vertical axis measures the mean number of branches explored by Korf's algorithm to decide if a coalition of k agents can manipulate a veto election. The horizontal axis measures $\log_2(w)/k$.

that the manipulators wish to win, but the k manipulators have exactly twice the weight of vetoes of the n agents. This election is finely balanced. The preferred candidate of the manipulators wins if and only if the manipulators perfectly partition their vetoes between the two candidates that they wish to lose. Note that this is precisely the trick used in reducing number partitioning to the manipulation problem by Conitzer et al. (2007). In Figure 20, we plot the probability that the k manipulators can make their preferred candidate win in such a "hung" election as we vary the size of their weights w. Similar to number partitioning (Gent & Walsh, 1998), we see a rapid transition in manipulability around $\log_2(w)/k \approx 1$. In Figure 21, we observe that there is a rapid increase in the computationally complexity to compute a manipulation around this point.

What happens when the votes are less correlated? We consider an election which is perfectly hung as before except for one agent who vetoes at random one of the three candidates. In Figure 22, we plot the cost of computing a manipulation as the weight of this single random veto increases. Even one uncorrelated vote is enough to make manipulation easy if it has the same magnitude in weight as the vetoes of the manipulators. This suggests that we will find hard manipulation problems in veto elections only when votes are highly correlated.

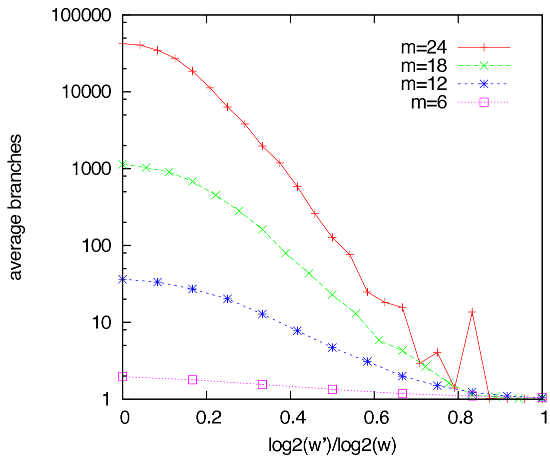

Figure 22: The impact of one random agent on the manipulability of a "hung" veto election. Vetoes of the manipulators are weighted and weights are uniformly drawn from $(0, w]$, the non-manipulating agents have all vetoed the candidate that the manipulators wish to win, and the sum of the weights of the manipulators is twice that of the non-manipulators except for one random non-manipulating agent whose weight is uniformly drawn from $(0, w']$. The vertical axis measures the mean number of search branches explored by Korf's algorithm to decide if a coalition of k agents can manipulate a veto election. The horizontal axis measures $\log_2(w')/\log_2(w)$. When the veto of the one random agent has the same weight as the other agents, it is computationally easy to decide if the election can be manipulated.

10. Related Work

As indicated earlier, a number of theoretical results suggest elections are easy to manipulate in practice despite worst case NP-hardness results. For example, Procaccia and Rosenschein proved that for most scoring rules and a wide variety of distributions over votes, when the size of the coalition is $o(\sqrt{n})$, the probability that they can change the result tends to 0, and when it is $\omega(\sqrt{n})$, the probability that they can manipulate the result tends to 1 (Procaccia & Rosenschein, 2007a). They also gave a simple greedy procedure that will find a manipulation of a scoring rule in polynomial time with a probability of failure that is an inverse polynomial in n (Procaccia & Rosenschein, 2007b). However, we should treat this result with caution as the "junta" distributions used in this work may be of limited usefulness (Erdélyi, Hemaspaandra, Rothe, & Spakowski, 2009).

As a second example, Xia and Conitzer have shown that for a large class of voting rules including STV, as the number of agents grows, either the probability that a coalition can manipulate the result is very small (as the coalition is too small), or the probability that they can easily manipulate the result to make any alternative win is very large (Xia & Conitzer, 2008a). They left open only a small interval in the size for the coalition for which the coalition is large enough to manipulate but not obviously large enough to manipulate the result easily.

Friedgut, Kalai and Nisan proved that if the voting rule is neutral and far from dictatorial and there are 3 candidates then there exists an agent for whom a random manipulation succeeds with probability $\Omega(\frac{1}{n})$ (Friedgut et al., 2008). They were, however, unable to extend their proof to four (or more) candidates. More recently, Isaksson, Kindler and Mossel proved a similar result for 4 or more candidates using geometric arguments (Isaksson, Kindler, & Mossel, 2010). Starting from different assumptions again, Xia and Conitzer showed that a random manipulation would succeed with probability $\Omega(\frac{1}{n})$ for 3 or more candidates for STV, for 4 or more candidates for any scoring rule and for 5 or more candidates for Copeland (Xia & Conitzer, 2008b).

As discussed earlier, Coleman and Teague proposed some algorithms to compute manipulations for the STV rule (Coleman & Teague, 2007). They also conducted an empirical study which demonstrated that only relatively small coalitions are needed to change the elimination order of the STV rule. They observed that most uniform and random elections are not trivially manipulable using a simple greedy heuristic. On the other hand, our results suggest that, for manipulation by a single agent, often only a small amount of backtracking is needed to find a manipulation or prove that none exists.

11. Conclusions

We have studied empirically whether computational complexity is a barrier to the manipulation for the STV and veto rules when the manipulating agents have complete information about the other votes. We have looked at a number of different distributions of votes including uniform random votes, correlated votes drawn from an urn model, and votes sampled from some real world elections. We have looked at manipulation by a single unweighted agent in the case of STV, and by a coalition of weighted agents in the case of veto voting. In many of the elections in our experiments, it was easy to compute a manipulation or to prove no manipulation was possible. The situations we identified where manipulations were computationally difficult to find depended either on the election having hundreds of candidates or on the election being very tightly hung. These results increase the concern that computational complexity may not be a significant barrier to manipulation in practice.

What other lessons can be learnt from this study? First, whilst we have focused on the STV and veto rules, similar behavior is likely with other voting rules. It would, for instance, be interesting to study the Borda rule as this is one of the few rules used in practice where computing a manipulation is NP-hard with unweighted votes (Davies et al., 2011; Betzler et al., 2011). It would also be interesting to study voting rules like Copeland, maximin and ranked pairs. These rules are members of the small set of voting rules which are NP-hard to manipulate without weights on the votes (Xia, Zuckerman, Procaccia, Conitzer, & Rosenschein, 2009). Second, there may be a connection between the smoothness of the phase transition and problem hardness. Sharp phase transitions like that for propositional satisfiability are associated with hard decision problems, whilst smooth transitions are associated with easy instances of NP-hard problems and with polynomial problems like 2-colourability. The phase transitions observed here appear to be smooth. Third, given the insights provided by empirical studies, it would be interesting to consider similar studies for related problems. For example, is computational complexity an issue in preference elicitation (Conitzer & Sandholm, 2002b; Walsh, 2008; Pini, Rossi, Venable, & Walsh, 2008)? Fourth, we have assumed that the manipulators have complete information about the votes of the other agents. It is an interesting future direction to determine if uncertainty in how agents have voted adds to the computational complexity of manipulation in practice (Conitzer & Sandholm, 2002a; Walsh, 2007; Lang, Pini, Rossi, Venable, & Walsh, 2007).

Acknowledgments

NICTA is funded by the Australian Government through the Department of Broadband, Communications and the Digital Economy and the Australian Research Council through the ICT Centre of Excellence program. Some of the results in this paper appeared in two earlier conference papers (Walsh, 2009, 2010).

References

Achlioptas, D. (1999). *Threshold phenomena in random graph colouring and satisfiability*. Ph.D. thesis, Department of Computer Science, University of Toronto.

Achlioptas, D., Chtcherba, A., Istrate, G., & Moore, C. (2001). The phase transition in 1-in-k SAT and NAE SAT. In *Proceedings of the 12th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA'01)*, pp. 719–720. Society for Industrial and Applied Mathematics.

Bailey, D., Dalmau, V., & Kolaitis, P. (2001). Phase transitions of PP-complete satisfiability problems. In Nebel, B. (Ed.), *Proceedings of 17th International Joint Conference on Artificial Intelligence (IJCAI 2001)*, pp. 183–189. International Joint Conference on Artificial Intelligence, Morgan Kaufmann.

Barberà, S., Berga, D., & Moreno, B. (2009). Single-dipped preferences. UFAE and IAE working papers 801.09, Unitat de Fonaments de l'Anlisi Econmica (UAB) and Institut d'Anlisi Econmica (CSIC).

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

Bartholdi, J., Tovey, C., & Trick, M. (1989). The computational difficulty of manipulating an election. *Social Choice and Welfare*, *6*(3), 227–241.

Berg, S. (1985). Paradox of voting under an urn model: the effect of homogeneity. *Public Choice*, *47*, 377–387.

Betzler, N., Niedermeier, R., & Woeginger, G. (2011). Unweighted coalitional manipulation under the Borda rule is NP-hard. In Walsh, T. (Ed.), *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI 2011)*. International Joint Conference on Artificial Intelligence.

Black, D. (1948). On the rationale of group decision-making. *Journal of Political Economy*, *56*(1), 23–34.

Brandt, F., Brill, M., Hemaspaandra, E., & Hemaspaandra, L. (2010). Bypassing combinatorial protections: Polynomial-time algorithms for single-peaked electorates. In Fox, M., & Poole, D. (Eds.), *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI 2010)*. AAAI Press.

Chamberlin, J. (1985). An investigation into the relative manipulability of four voting systems. *Behavioral Science*, *30*, 195–203.

Cheeseman, P., Kanefsky, B., & Taylor, W. (1991). Where the really hard problems are. In Mylopoulos, J., & Reiter, R. (Eds.), *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI 1991)*, pp. 331–337. International Joint Conference on Artificial Intelligence.

Coleman, T., & Teague, V. (2007). On the complexity of manipulating elections. In Gudmundsson, J., & Jay, B. (Eds.), *Proceedings of the 13th Australasian Symposium on Theory of Computing (CATS '07)*, pp. 25–33. Australian Computer Society, Inc.

Conitzer, V. (2007). Eliciting single-peaked preferences using comparison queries. In Durfee, E., Yokoo, M., Huhns, M., & Shehory, O. (Eds.), *Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2007)*, pp. 408–415. IFAAMAS.

Conitzer, V. (2009). Eliciting single-peaked preferences using comparison queries. *Journal of Artificial Intelligence Research*, *35*, 161–191.

Conitzer, V., & Sandholm, T. (2002a). Complexity of manipulating elections with few candidates. In Dechter, R., Kearns, M., & Sutton, R. (Eds.), *Proceedings of the 18th National Conference on Artificial Intelligence (AAAI 2002)*, pp. 314–319. Association for Advancement of Artificial Intelligence.

Conitzer, V., & Sandholm, T. (2002b). Vote elicitation: Complexity and strategy-proofness. In Dechter, R., Kearns, M., & Sutton, R. (Eds.), *Proceedings of the 18th National Conference on Artificial Intelligence (AAAI 2002)*, pp. 392–397. Association for Advancement of Artificial Intelligence.

Conitzer, V., & Sandholm, T. (2006). Nonexistence of voting rules that are usually hard to manipulate. In Gil, Y., & Mooney, R. (Eds.), *Proceedings of the 21st National Conference on Artifical Intelligence (AAAI 2006)*, pp. 627–634. Association for Advancement of Artificial Intelligence.

Conitzer, V., Sandholm, T., & Lang, J. (2007). When are elections with few candidates hard to manipulate?. *Journal of the Association for Computing Machinery*, *54*(3). Article 14 (33 pages).

Davies, J., Katsirelos, G., Narodytska, N., & Walsh, T. (2011). Complexity of and algorithms for Borda manipulation. In Burgard, W., & Roth, D. (Eds.), *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence (AAAI 2011)*. AAAI Press.

Dobra, J. (1983). An approach to empirical studies of voting paradoxes: An update and extension.. *Public Choice*, *41*, 241–250.

Dubois, O., Monasson, R., Selman, B., & Zecchina, R. (2001). Special issue: Phase transitions in combinatorial problems. *Theoretical Computer Science*, *265*(1–2), 1–306.

Dyer, J., & Miles, R. (1976). An actual application of collective choice theory to the selection of trajectories for the Mariner Jupiter/Saturn 1977 project. *Operations Research*, *24*(2), 220–244.

Erdélyi, G., Hemaspaandra, L., Rothe, J., & Spakowski, H. (2009). Generalized Juntas and NP-hard sets. *Theoretical Computer Science*, *410*(38-40), 3995–4000.

Faliszewski, P., Hemaspaandra, E., & Hemaspaandra, L. (2010). Using complexity to protect elections. *Communications of the ACM*, *53*(11), 74–82.

Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L., & Rothe, J. (2009). The shield that never was: societies with single-peaked preferences are more open to manipulation and control. In Heifetz, A. (Ed.), *Proceedings of the 12th Conference on Theoretical Aspects of Rationality and Knowledge (TARK-2009)*, pp. 118–127.

Faliszewski, P., Hemaspaandra, E., & Schnoor, H. (2008). Copeland voting: ties matter. In Padgham, L., Parkes, D., Müller, J., & Parsons, S. (Eds.), *7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, pp. 983–990.

Faliszewski, P., & Procaccia, A. (2010). AI's war on manipulation: Are we winning?. *AI Magazine*, *31*(4), 53–64.

Friedgut, E., Kalai, G., & Nisan, N. (2008). Elections can be manipulated often. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2008)*, pp. 243–249. IEEE Computer Society Press.

Gent, I., Hoos, H., Prosser, P., & Walsh, T. (1999). Morphing: Combining structure and randomness. In Hendler, J., & Subramanian, D. (Eds.), *Proceedings of the 16th National Conference on Artificial Intelligence (AAAI 1999)*, pp. 654–660. Association for Advancement of Artificial Intelligence.

Gent, I., MacIntyre, E., Prosser, P., Smith, B., & Walsh, T. (2001). Random constraint satisfaction: Flaws and structure. *Constraints*, *6*(4), 345–372.

Gent, I., MacIntyre, E., Prosser, P., & Walsh, T. (1995). Scaling effects in the CSP phase transition. In Montanari, U., & Rossi, F. (Eds.), *Proceedings of the 1st International Conference on Principles and Practices of Constraint Programming (CP-95)*, Vol. 976 of *Lecture Notes in Computer Science*, pp. 70–87. Springer-Verlag.

Gent, I., & Walsh, T. (1994). The SAT phase transition. In Cohn, A. (Ed.), *Proceedings of the 11th European Conference on Artificial Intelligence (ECAI-94)*, pp. 105–109. John Wiley & Sons.

Gent, I., & Walsh, T. (1995). Phase transitions from real computational problems. In *Proceedings of the 8th International Symposium on Artificial Intelligence: Intelligent Systems Applications in Industry and Business*, pp. 356–364.

Gent, I., & Walsh, T. (1996a). Phase transitions and annealed theories: Number partitioning as a case study. In Wahlster, W. (Ed.), *Proc. of the 12th European Conference on Artificial Intelligence (ECAI-96)*, pp. 170–174. John Wiley and Sons, Chichester.

Gent, I., & Walsh, T. (1996b). The satisfiability constraint gap. *Artificial Intelligence*, *81*(1–2), 59–80.

Gent, I., & Walsh, T. (1996c). The TSP phase transition. *Artificial Intelligence*, *88*, 349–358.

Gent, I., & Walsh, T. (1998). Analysis of heuristics for number partitioning. *Computational Intelligence*, *14*(3), 430–451.

Gent, I., & Walsh, T. (1999). Beyond NP: the QSAT phase transition. In Hendler, J., & Subramanian, D. (Eds.), *Proceedings of the 16th National Conference on AI*, pp. 648–653. Association for Advancement of Artificial Intelligence.

Gibbard, A. (1973). Manipulation of voting schemes: A general result. *Econometrica*, *41*, 587–601.

Gomes, G., & Walsh, T. (2006). Randomness and structure. In Rossi, F., van Beek, P., & Walsh, T. (Eds.), *Handbook of Constraint Programming*, Foundations of Artificial Intelligence, pp. 639–664. Elsevier.

Hartmann, A., & Weigt, M. (2005). *Phase Transitions in Combinatorial Optimization Problems: Basics, Algorithms and Statistical Mechanics*. Wiley-VCH, Weinheim.

Isaksson, M., Kindler, G., & Mossel, E. (2010). The geometry of manipulation: A quantitative proof of the Gibbard-Satterthwaite theorem. In *51th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2010)*, pp. 319–328. IEEE Computer Society.

Karmarkar, N., Karp, R., Lueker, J., & Odlyzko, A. (1986). Probabilistic analysis of optimum partitioning. *Journal of Applied Probability*, *23*, 626–645.

Korf, R. (1995). From approximate to optimal solutions: A case study of number partitioning. In Mellish, C. S. (Ed.), *Proceedings of the 14th International Joint Conference on Artificial Intelligence (IJCAI 1995)*, pp. 266–272. International Joint Conference on Artificial Intelligence.

Lang, J., Pini, M., Rossi, F., Venable, B., & Walsh, T. (2007). Winner determination in sequential majority voting. In Veloso, M. M. (Ed.), *Proceedings of the 20th International Joint Conference on Artificial Itelligence (IJCAI-2007)*, pp. 1372–1377. International Joint Conference on Artificial Intelligence.

McCabe-Dansted, J., & Slinko, A. (2006). Exploratory analysis of similarities between social choice rules. *Group Decision and Negotiation*, *15*, 77–107.

Mertens, S. (2001). A physicist's approach to number partitioning. *Theoretical Computer Science*, *265*(1-2), 79–108.

Mitchell, D., Selman, B., & Levesque, H. (1992). Hard and Easy Distributions of SAT Problems. In *Proceedings of the 10th National Conference on AI*, pp. 459–465. Association for Advancement of Artificial Intelligence.

Pini, M., Rossi, F., Venable, K., & Walsh, T. (2008). Dealing with incomplete agents' preferences and an uncertain agenda in group decision making via sequential majority voting. In Brewka, G., & Lang, J. (Eds.), *Principles of Knowledge Representation and Reasoning: Proceedings of the Eleventh International Conference (KR 2008)*, pp. 571–578. AAAI Press.

Procaccia, A. D., & Rosenschein, J. S. (2007a). Average-case tractability of manipulation in voting via the fraction of manipulators. In Durfee, E. H., Yokoo, M., Huhns, M. N., & Shehory, O. (Eds.), *Proceedings of 6th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-07)*, pp. 718–720. IFAAMAS.

Procaccia, A. D., & Rosenschein, J. S. (2007b). Junta distributions and the average-case complexity of manipulating elections. *Journal of Artificial Intelligence Research*, *28*, 157–181.

Prosser, P. (1994). Binary constraint satisfaction problems: Some are harder than others. In Cohn, A. G. (Ed.), *Proceedings of the 11th European Conference on Artificial Intelligence*, pp. 95–99. European Conference on Artificial Intelligence, John Wiley and Sons.

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*, 187–216.

Slaney, J., & Walsh, T. (2002). Phase transition behavior: from decision to optimization. In *Proceedings of the 5th International Symposium on the Theory and Applications of Satisfiability Testing, SAT 2002*.

Slinko, A., & White, S. (2008). Non- dictatorial social choice rules are safely manipulable. In Goldberg, U. E. . P. W. (Ed.), *Proceedings of 2nd International Workshop on Computational Social Choice (COMSOC'08)*, pp. 403–413.

Smith, B. (1994). The phase transition in constraint satisfaction problems: A closer look at the mushy region. In Cohn, A. G. (Ed.), *Proceedings of the 11th European Conference on Artificial Intelligence*, pp. 100–104. European Conference on Artificial Intelligence, John Wiley and Sons.

Tideman, T. (1987). Independence of clones as a criterion for voting rules. *Social Choice and Welfare*, *4*, 185–206.

Walsh, T. (1998). The constrainedness knife-edge. In Mostow, J., & Rich, C. (Eds.), *Proceedings of the 15th National Conference on AI*, pp. 406–411. Association for Advancement of Artificial Intelligence.

Walsh, T. (1999). Search in a small world. In Dean, T. (Ed.), *Proceedings of the 16th International Joint Conference on Artificial Itelligence (IJCAI-99)*, pp. 1172–1177. International Joint Conference on Artificial Intelligence, Morgan Kaufmann.

Walsh, T. (2001). Search on high degree graphs. In Nebel, B. (Ed.), *Proceedings of the 17th International Joint Conference on Artificial Itelligence (IJCAI-2001)*, pp. 266–274. International Joint Conference on Artificial Intelligence, Morgan Kaufmann.

Walsh, T. (2002). From P to NP: COL, XOR, NAE, 1-in-k, and Horn SAT. In Dechter, R., Kearns, M., & Sutton, R. (Eds.), *Proceedings of the 17th National Conference on AI (AAAI 2002)*, pp. 695–700. Association for Advancement of Artificial Intelligence.

Walsh, T. (2007). Uncertainty in preference elicitation and aggregation. In *Proceedings of the 22nd National Conference on AI*, pp. 3–8. Association for Advancement of Artificial Intelligence.

Walsh, T. (2008). Complexity of terminating preference elicitation. In Padgham, L., Parkes, D. C., Müller, J. P., & Parsons, S. (Eds.), *7th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2008)*, pp. 967–974. IFAAMAS.

Walsh, T. (2009). Where are the really hard manipulation problems? The phase transition in manipulating the veto rule. In Boutilier, C. (Ed.), *Proceedings of the 21st International Joint Conference on Artificial Itelligence (IJCAI-2009)*, pp. 324–329. International Joint Conference on Artificial Intelligence.

Walsh, T. (2010). An empirical study of the manipulability of single transferable voting. In Coelho, H., Studer, R., & Wooldridge, M. (Eds.), *Proc. of the 19th European Conference on Artificial Intelligence (ECAI-2010)*, Vol. 215 of *Frontiers in Artificial Intelligence and Applications*, pp. 257–262. IOS Press.

Xia, L., & Conitzer, V. (2008a). Generalized scoring rules and the frequency of coalitional manipulability. In Fortnow, L., Riedl, J., & Sandholm, T. (Eds.), *EC '08: Proceedings of the 9th ACM conference on Electronic Commerce*, pp. 109–118. ACM.

Xia, L., & Conitzer, V. (2008b). A sufficient condition for voting rules to be frequently manipulable. In Fortnow, L., Riedl, J., & Sandholm, T. (Eds.), *Proceedings of the 9th ACM conference on Electronic Commerce (EC' 08)*, pp. 99–108. ACM.

Xia, L., Zuckerman, M., Procaccia, A., Conitzer, V., & Rosenschein, J. (2009). Complexity of unweighted coalitional manipulation under some common voting rules. In Boutilier, C. (Ed.), *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI 2009)*, pp. 348–353. International Joint Conference on Artificial Intelligence.

Zhang, W., & Korf, R. (1996). A study of complexity transitions on the asymmetic traveling salesman problem. *Artificial Intelligence*, *81*(1-2), 223–239.