

Is Computational Complexity a Barrier to Manipulation?

Toby Walsh

NICTA and University of NSW, Sydney, Australia, email: toby.walsh@nicta.com.au

Abstract. When agents are acting together, they may need a simple mechanism to decide on joint actions. One possibility is to have the agents express their preferences in the form of a ballot and use a voting rule to decide the winning action(s). Unfortunately, agents may try to manipulate such an election by misreporting their preferences. Fortunately, 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. To address this issue, I suggest studying empirically if computational complexity is in practice a barrier to manipulation. The basic tool used in my investigations is the identification of computational “phase transitions”. Such an approach has been fruitful in identifying hard instances of propositional satisfiability and other NP-hard problems. I show that phase transition behaviour gives insight into the hardness of manipulating voting rules, increasing concern that computational complexity is indeed any sort of barrier. Finally, I look at the problem of computing manipulation of other, related problems like stable marriage and tournament problems.

1 Introduction

The Gibbard Satterthwaite theorem proves that, under some simple assumptions, a voting rule can always be manipulated. In an influential paper [1], Bartholdi, Tovey and Trick proposed an appealing escape: perhaps it is computationally so difficult to find a successful manipulation that agents have little option but to report their true preferences? To illustrate this idea, they demonstrated that the second order Copeland rule is NP-hard to manipulate. Shortly after, Bartholdi and Orlin proved that the more well known Single Transferable Voting (STV) rule is NP-hard to manipulate [2]. Many other voting rules have subsequently been proven to be NP-hard to manipulate [3]. There is, however, increasing concern that worst-case results like these do not reflect the difficulty of manipulation in practice. Indeed, several theoretical results suggest that manipulation may often be easy (e.g. [4]).

2 Empirical analysis

In addition to attacking this question theoretically, I have argued in a recent series of papers that we may benefit from studying it empirically [5, 6]. There are several reasons why empirical analysis is useful. First, theoretical analysis is often restricted to

particular distributions like uniform votes. Manipulation may be very different in practice due to correlations in the preferences of the agents. For instance, if all preferences are single-peaked then there voting rules where it is in the best interests of all agents to state their true preferences. Second, theoretical analysis is often asymptotic so does not reveal the size of hidden constants. The size of such constants may be important to the actual computational cost of computing a manipulation. In addition, elections are typically bounded in size. Is asymptotic behaviour relevant to the size of elections met in practice? An empirical study may quickly suggest if the result extends to more candidates. Finally, empirical studies can suggest theorems to prove. For instance, our experiments suggest a simple formula for the probability that a coalition is able to elect a desired candidate. It would be interesting to derive this exactly.

3 Voting rules

My empirical studies have focused on two voting rules: single transferable voting (STV) and veto voting. STV is representative of voting rules that are NP-hard to manipulate without weights on votes. Indeed, as I argue shortly, it is one of the few such rules. Veto voting is, on the other hand, a simple representative of rules where manipulation is NP-hard when votes are weighted or (equivalently) we have uncertainty about how agents have voted. The two voting rules therefore cover the two different cases where computational complexity has been proposed as a barrier to manipulation.

STV proceeds in a number of rounds. Each agent totally ranks the candidates on a ballot. Until one candidate has a majority of first place votes, we eliminate the candidate with the least number of first place votes. Ballots placing the eliminated candidate in first place are then re-assigned to the second place candidate. STV is used in a wide variety of elections including for the Australian House of Representatives, the Academy awards, and many organizations including the American Political Science Association, and the International Olympic Committee. STV has played a central role in the study of the computational complexity of manipulation. Bartholdi and Orlin argued that:

“STV is apparently unique among voting schemes in actual use today in that it is computationally resistant to manipulation.” (page 341 of [2]).

By comparison, the veto rule is a much simpler scoring rule in which each agent gets to cast a veto against one candidate. The candidate with the fewest vetoes wins. There are several reasons why the veto rule is interesting to study. The veto rule is very simple to reason about. This can be contrasted with other voting rules like STV. Part of the complexity of manipulating the STV rule appears to come from reasoning about what happens between the different rounds. The veto rule, on the other hand, has a single round. The veto rule is also on the borderline of tractability since constructive manipulation (that is, ensuring a particular candidate wins) of the veto rule by a coalition of weighted agents is NP-hard but destructive manipulation (that is, ensuring a particular candidate does not win) is polynomial [3].

4 Voting distributions

Empirical analysis requires collections of votes on which to compute manipulations. My analysis starts with one of the simplest possible scenarios: elections in which each vote is equally likely. We have one agent trying to manipulate an election of m candidates in which n other agents vote. Votes are drawn uniformly at random from all $m!$ possible votes. This is the Impartial Culture (IC) model. In many real life situations, however, votes are correlated with each other. I therefore also considered single-peaked preferences, single-troughed preferences, and votes drawn from the Polya Eggenberger urn model [7]. In an urn model, we have an urn containing all possible votes. We draw votes out of the urn at random, and put them back into the urn with a additional votes of the same type (where a is a parameter). This generalizes both the Impartial Culture model ($a = 0$) and the Impartial Anonymous Culture ($a = 1$) model. Real world elections may differ from these ensembles. I therefore also sampled some real voting records [8, 9]. Finally, one agent on their own is often unable to manipulate the result. I therefore also considered coalitions of agents who are trying to manipulate elections.

5 Results

My experiments suggest different behaviour occurs in the problem of computing manipulations of voting rules than in other NP-hard problems like propositional satisfiability and graph colouring [10, 11]. For instance, we often did not see a rapid transition that sharpens around a fixed point as in satisfiability [12]. Many transitions appear smooth and do not sharpen towards a step function as problem size increases. Such smooth phase transitions have been previously seen in polynomial problems [13]. In addition, hard instances often did not occur around some critical parameter. Figures 1 to 3 reproduce some typical graphs from [6].

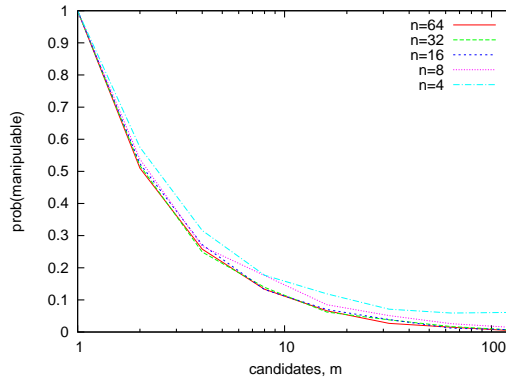


Fig. 1. Manipulability of correlated votes. The number of agents n is fixed and we vary the number of candidates m . The y-axis measures the probability that the manipulator can make a random candidate win.

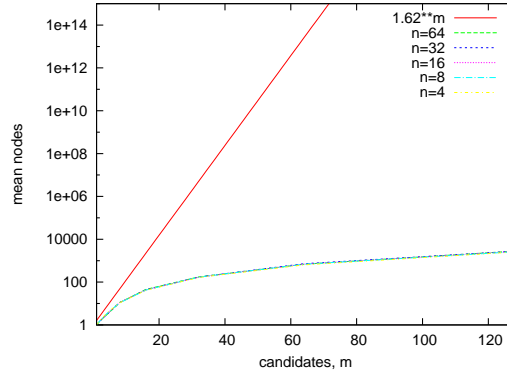


Fig. 2. Search to compute if an agent can manipulate an election with correlated votes. The number of agents n is fixed and we vary the number of candidates m . The y-axis measures the mean number of search nodes explored to compute a manipulation or prove that none exists. Median and other percentiles are similar. 1.62^m is the published worst-case bound for the recursive algorithm used to compute a manipulation [3].

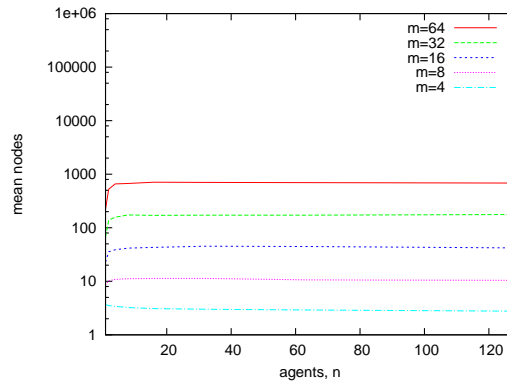


Fig. 3. Search to compute if an agent can manipulate an election with correlated votes. The number of candidates m is fixed and we vary the number of agents n .

Similar phase transition studies have been used to identify hard instances of NP-hard problems like propositional satisfiability [12, 17, 18], constraint satisfaction [19–23], number partitioning [24–26], Hamiltonian circuit [27, 28], and the traveling salesperson problem [29, 30]. Phase transition studies have also been used to study polynomial problems [31, 32] as well as higher complexity classes [33, 34] and optimization problems [35, 36]. Finally, phase transition studies have been used to study problem structure like small worldiness [37] and high degree nodes [38].

6 Other manipulation problems

Another multi-agent problem in which manipulation may be an issue is the stable marriage problem. This is the well-known problem of matching men to women so that no man and woman who are not married to each other both prefer each other. It has a wide variety of practical applications such as a matching doctors to hospitals. As with voting, an important issue is whether agents can manipulate the result by mis-reporting their preferences. Unfortunately, Roth [14] proved that *all* stable marriage procedures can be manipulated. We might hope that computational complexity might also be a barrier to manipulate stable marriage procedures. In joint work with Pini, Rossi and Venable, I have proposed a new stable marriage procedures that is NP-hard to manipulate [15]. Another advantage of this new procedure is that, unlike the Gale-Shapley algorithm, it does not favour one sex over the other. Our procedure picks the stable matching that is most preferred by the most popular men and women. The most preferred men and women are chosen using a voting rule. We prove that, if the voting rule used is STV then the resulting stable matching procedure is also NP-hard to manipulate. We conjecture that other voting rules which are NP-hard to manipulate will give rise to stable matching procedures which are also NP-hard to manipulate.

The final domain in which I have studied computational issues surrounding manipulation is that of (sporting) tournaments (joint work with Russell) [16]. Manipulating a tournament is slightly different to manipulating an election. In a sporting tournament, the voters are also the candidates. Since it is hard (without bribery or similar mechanisms) for a team to play better than it can, we consider just manipulations where the manipulators can throw games. We show that we can decide how to manipulate round robin and cup competitions, two of the most popular sporting competitions in polynomial time. In addition, we show that finding the minimal number of games that need to be thrown to manipulate the result can also be determined in polynomial time. Finally, we give a polynomial time procedure to calculate the probability that a team wins a cup competition under manipulation.

7 Conclusions

I have argued that empirical studies can provide insight into whether computational complexity is a barrier to the manipulation. Somewhat surprisingly, almost every one of the many millions of elections in the experiments in [5, 6] was easy to manipulate or to prove could not be manipulated. Such experimental results increase the concerns that computational complexity is indeed a barrier to manipulation in practice. Many other

voting rules have been proposed which could be studied in the future. Two interesting rules are maximin and ranked pairs. These two rules have only recently been shown to be NP-hard to manipulate, and are members of the small set of voting rules which are NP-hard to manipulate without weights or uncertainty [39]. These results demonstrate that empirical studies can provide insight into the computational complexity of computing manipulations. It would be interesting to consider similar phase transition studies for related problems like preference elicitation [40, 41].

Acknowledgements: NICTA is funded by the Department of Broadband, Communications and the Digital Economy, and the Australian Research Council.

References

1. Bartholdi, J., Tovey, C., Trick, M.: The computational difficulty of manipulating an election. *Social Choice and Welfare* **6** (1989) 227–241
2. Bartholdi, J., Orlin, J.: Single transferable vote resists strategic voting. *Social Choice and Welfare* **8** (1991) 341–354
3. Conitzer, V., Sandholm, T., Lang, J.: When are elections with few candidates hard to manipulate. *Journal of the Association for Computing Machinery* **54** (2007)
4. Xia, L., Conitzer, V.: A sufficient condition for voting rules to be frequently manipulable. In: *EC '08: Proc. of the 9th ACM conference on Electronic commerce*, New York, NY, USA, ACM (2008) 99–108
5. Walsh, T.: Where are the really hard manipulation problems? The phase transition in manipulating the veto rule. In: *Proc. of 21st IJCAI, Int. Joint Conf. on Artificial Intelligence* (2009) 324–329
6. Walsh, T.: An empirical study of the manipulability of single transferable voting. In: *Proc. of the 19th ECAI*, IOS Press (2010)
7. Berg, S.: Paradox of voting under an urn model: the effect of homogeneity. *Public Choice* **47** (1985) 377–387
8. Gent, I., Walsh, T.: Phase transitions from real computational problems. In: *Proc. of the 8th Int. Symposium on Artificial Intelligence*. (1995) 356–364
9. Gent, I., Hoos, H., Prosser, P., Walsh, T.: Morphing: Combining structure and randomness. In: *Proc. of the 16th National Conf. on AI, AAAI* (1999)
10. Cheeseman, P., Kanefsky, B., Taylor, W.: Where the really hard problems are. In: *Proc. of the 12th IJCAI, Int. Joint Conf. on Artificial Intelligence* (1991) 331–337
11. Walsh, T.: The constrainedness knife-edge. In: *Proc. of the 15th National Conf. on AI, AAAI* (1998)
12. Mitchell, D., Selman, B., Levesque, H.: Hard and Easy Distributions of SAT Problems. In: *Proc. of the 10th National Conf. on AI, AAAI* (1992) 459–465
13. Walsh, T.: From P to NP: COL, XOR, NAE, 1-in-k, and Horn SAT. In: *Proc. of the 17th National Conf. on AI, AAAI* (2002)
14. Roth, A.E.: The economics of matching: Stability and incentives. *Mathematics of Operations Research* **7** (1982) 617–628
15. Pini, M., Rossi, F., Venable, K., T. W.: Manipulation and gender neutrality in stable marriage procedures. In: *8th Int. Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2009)*. (2009) 665–672
16. Russell, T., Walsh, T.: Manipulating tournaments in cup and round robin competitions. In: Rossi, F., Tsoukiàs, A., eds.: *Algorithmic Decision Theory, First Int. Conf. (ADT 2009)*, (2009) 26–37

17. Gent, I., Walsh, T.: The SAT phase transition. In Cohn, A.G., ed.: Proc. of 11th ECAI, John Wiley & Sons (1994) 105–109
18. Gent, I., Walsh, T.: The satisfiability constraint gap. *Artificial Intelligence* **81** (1996)
19. Prosser, P.: Binary constraint satisfaction problems: Some are harder than others. In: Proc. of the 11th ECAI, (1994) 95–99
20. Smith, B.: The phase transition in constraint satisfaction problems: A closer look at the mushy region. In: Proc. of the 11th ECAI, (1994)
21. Gent, I., MacIntyre, E., Prosser, P., Walsh, T.: Scaling effects in the CSP phase transition. In: 1st Int. Conf. on Principles and Practices of Constraint Programming (CP-95), (1995) 70–87
22. Gent, I., MacIntyre, E., Prosser, P., Walsh, T.: The constrainedness of search. In: Proc. of the 13th National Conf. on AI, AAAI (1996) 246–252
23. Gent, I., MacIntyre, E., Prosser, P., Smith, B., Walsh, T.: Random constraint satisfaction: Flaws and structure. *Constraints* **6** (2001) 345–372
24. Mertens, S.: A physicist's approach to number partitioning. *Theoretical Computer Science* **265** (2001) 79–108
25. Gent, I., Walsh, T.: Phase transitions and annealed theories: Number partitioning as a case study. In: Proc. of 12th ECAI. (1996)
26. Gent, I., Walsh, T.: Analysis of heuristics for number partitioning. *Computational Intelligence* **14** (1998) 430–451
27. Frank, J., Gent, I., Walsh, T.: Asymptotic and finite size parameters for phase transitions: Hamiltonian circuit as a case study. *Information Processing Letters* **66** (1998) 241–245
28. Vandegriend, B., Culberson, J.: The Gn,m phase transition is not hard for the Hamiltonian Cycle problem. *Journal of Artificial Intelligence Research* **9** (1998) 219–245
29. Gent, I., Walsh, T.: The TSP phase transition. *Artificial Intelligence* **88** (1996) 349–358
30. Zhang, W.: Phase transitions and backbones of the asymmetric traveling salesman problem. *Journal of Artificial Intelligence Research* **21** (2004) 471–497
31. Grant, S., Smith, B.: The phase transition behaviour of maintaining arc consistency. In Wahlster, W., ed.: Proc. of the 12th ECAI, Wiley (1996) 175–179
32. Gent, I., MacIntyre, E., Prosser, P., Shaw, P., Walsh, T.: The constrainedness of arc consistency. In: 3rd Int. Conf. on Principles and Practices of Constraint Programming (CP-97), (1997) 327–340
33. Gent, I., Walsh, T.: Beyond NP: the QSAT phase transition. In: Proc. of the 16th National Conf. on AI, AAAI (1999)
34. Bailey, D., Dalmau, V., Kolaitis, P.: Phase transitions of PP-complete satisfiability problems. In: Proc. of the 17th IJCAI, Int. Joint Conf. on Artificial Intelligence (2001) 183–189
35. Slaney, J., Walsh, T.: Phase transition behavior: from decision to optimization. In: Proc. of the 5th Int. Symposium on the Theory and Applications of Satisfiability Testing, SAT 2002. (2002)
36. Larrosa, J., Meseguer, P.: Phase transition in MAX-CSP. In Wahlster, W., ed.: Proc. of the 12th ECAI, Wiley (1996) 190–194
37. Walsh, T.: Search in a small world. In: Proc. of 16th IJCAI, Int. Joint Conf. on Artificial Intelligence (1999)
38. Walsh, T.: Search on high degree graphs. In: Proc. of 17th IJCAI, Int. Joint Conf. on Artificial Intelligence (2001)
39. Xia, L., Zuckerman, M., Procaccia, A., Conitzer, V., Rosenschein, J.: Complexity of unweighted coalitional manipulation under some common voting rules. In: Proc. of 21st IJCAI, Int. Joint Conf. on Artificial Intelligence (2009) 348–353
40. Walsh, T.: Uncertainty in preference elicitation and aggregation. In: Proc. of the 22nd National Conf. on AI, AAAI (2007)
41. Walsh, T.: Complexity of terminating preference elicitation. In: 7th Int. Joint Conf. on Autonomous Agents and Multiagent Systems, IFAAMAS (2008)