Manipulation with Randomized Tie-Breaking under Maximin

(Extended Abstract)

Michael Zuckerman and Jeffrey S. Rosenschein
School of Computer Science and Engineering
The Hebrew University of Jerusalem, Israel
{michez, jeff}@cs.huji.ac.il

ABSTRACT

In recent papers, Obraztsova et al. initiated the study of the computational complexity of voting manipulation under randomized tie-breaking [3, 2]. The authors provided a polynomial-time algorithm for the problem of finding an optimal vote for the manipulator (a vote maximizing the manipulator's expected utility) under the Maximin voting rule, for the case where the manipulator's utilities of the candidates are given by the vector $(1, 0, \ldots, 0)$. On the other hand, they showed that this problem is NP-hard for the case where the utilities are $(1, \ldots, 1, 0)$.

This paper continues that line of research. We prove that when the manipulator's utilities of the candidates are given by the vector $(1, \ldots, 1, 0, \ldots, 0)$, with k 1's and $(m-k)$ 0's, then the problem of finding an optimal vote for the manipulator is fixed-parameter tractable when parameterized by k. Also, by exploring the properties of the graph built by the algorithm, we prove that when a certain sub-graph of this graph contains a 2-cycle, then the solution returned by the algorithm is optimal.

Categories and Subject Descriptors

I.2.11 [**Artificial Intelligence**]: Distributed Artificial Intelligence—*Multiagent Systems*

General Terms

Theory, Algorithms, Economics

Keywords

Computational Social Choice, Voting, Game Theory

1. INTRODUCTION

Social choice theory provides tools for formalizing preference aggregation among agents, using a wide variety of voting rules. The work of Gibbard and Satterthwaite [1, 4] showed, however, that with any reasonable voting rule, there would always be the possibility of a situation where agents were better off voting strategically, reporting untrue preferences to the voting mechanism in an attempt to manipulate the outcome. One of the popular techniques to overcome the susceptibility to manipulation uses computational complexity. Manipulation is always potentially useful, but in practice it might be exponentially difficult to find a useful manipulation (in the worst case). Complexity could potentially serve as a useful defense, as it does in cryptography.

Most recently, attention has been turned to the question of ties and tie-breaking rules; in a recent paper, Obraztsova et al. proposed an algorithm for finding an optimal vote under the Maximin voting rule when randomized tie-breaking is used, for a particular special case of manipulator utilities [3]. This work was extended later by a subset of the same authors [2], where they showed that for another special case of manipulator utilities, finding a "good enough" manipulation under Maximin is NP-complete. The bottom line of these papers is that "ties matter", i.e., the way in which ties are broken influences fundamental characteristics of voting rules, in particular their computationally feasible susceptibility to manipulation. The current paper continues this line of research, by more fully characterizing the nature of manipulation in the Maximin voting rule for more general settings of manipulator utilities.

2. PRELIMINARIES

Voting An election is given by a set $C = \{c_1, \ldots, c_m\}$ of *candidates* (also called *alternatives*), and a set $V = \{v_1, \ldots, v_n\}$ of *voters*. The voters submit linear orders, R_i, over the candidates. We will sometimes use $\succ_i$ instead of R_i, for readability. If $c_k \succ_i c_j$, we say that i prefers c_k to c_j. We denote by $\mathcal{L}(C)$ the set of all linear orders over C. A list of n linear orders $\mathcal{R} = (R_1, \ldots, R_n) \in \mathcal{L}(C)^n$ is called a *preference profile*.

A *voting correspondence* is a mapping $\mathcal{F} : \mathcal{L}(C)^n \to 2^C$ which for every profile of the votes R determines a non-empty set of winners $S \subseteq C$. If $|\mathcal{F}(R)| = 1$, $\mathcal{F}$ is called a *voting rule*. In order to transform a voting correspondence into a voting rule, we need a *tie-breaking rule*. Formally, a tie-breaking rule is a mapping $\mathcal{T}$ which, given a non-empty set of tied candidates S, returns the winning candidate $c \in S$. In this work, we use the *randomized tie-breaking rule*, i.e., the rule where ties are broken uniformly at random.

In this paper we consider the Maximin voting rule (or, more precisely, voting correspondence). The maximin score of a candidate $c \in C$ is defined as the number of voters who prefer c to c's toughest opponent, i.e., $\min_{d \in C \setminus \{c\}} |\{i \mid c \succ_i d\}|$. The candidates with maximum score win.

Manipulation: Given a preference profile $\mathcal{R} = (R_1, \ldots, R_n)$

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), 4-8 June 2012, Valencia, Spain.

over a set of candidates C, for any preference order $L \in \mathcal{L}(C)$ we denote by (R_{-i}, L) the profile $(R_1, \ldots, R_{i-1}, L, R_{i+1}, \ldots, R_n)$. In order to model the manipulation with randomized tie-breaking, we follow [3] and [2], and assume that the manipulator has non-negative utilities over the set of candidates, $u(c)$ for every $c \in C$. We assume that the utilities are consistent with the manipulator's preference order $\succ_i$, i.e., $u(a) \geq u(b)$ if and only if $a \succ_i b$. In this work, we deal with the case where for all $c \in C$, $u(c) \in \{0, 1\}$. Now, if a voting correspondence $\mathcal{F}$ outputs a set $S \subseteq C$, the manipulator's expected utility is $\hat{u}(S) = \frac{1}{|S|} \sum_{c \in S} u(c)$. L is said to be v_i's optimal vote if for all linear orders $L' \in \mathcal{L}(C)$ it holds that $\hat{u}(\mathcal{F}(R_{-i}, L)) \geq \hat{u}(\mathcal{F}(R_{-i}, L'))$.

3. RESULTS

3.1 Parameterized Complexity Result

The next theorem continues the line of research started by Obraztsova et al. [3, 2], providing the parameterized complexity of manipulation when the manipulator utilities are given by the vector $(1, \ldots, 1, 0, \ldots, 0)$ with k 1's.

Theorem 1. *Let $1 \leq k \leq m-1$. Suppose that the utilities of the manipulator are as follows: $u(c_i) = 1$ for $1 \leq i \leq k$, $u(c_i) = 0$ for $k+1 \leq i \leq m$, where the order c_i on the alternatives is the preference order of the manipulator. Then the problem of finding an optimal manipulation is in FPT (fixed-parameter tractable), when parameterized by k. More specifically, there exists an algorithm for finding an optimal manipulation in $O(k!k^2 + (n+m)m^2)$ time.*

Proof. We consider an election $E = (C, V)$ where $C = \{c_1, \ldots, c_m\}$, $V = \{v_1, \ldots, v_n\}$, and v_n is the manipulator. We denote for a candidate $c_i \in C$ by $s(c_i)$ the Maximin score of c_i in the election $E' = (C, V')$, where $V' = \{v_1, \ldots, v_{n-1}\}$. Let $s = \max_{c_i \in C}\{s(c_i)\}$. Suppose that the utilities of the manipulator are as defined above. Let $C_1 = \{c_1, \ldots, c_k\}$ be the set of candidates having utility 1, and $C_0 = C \setminus C_1$ be the set of candidates with utility 0. Let $X = \text{argmax}_{c_i \in C_1}\{s(c_i)\}$. Since the manipulator can only increase the score of any candidate by 1 or by 0, if for $x \in X$, $s(x) < s - 1$ then, clearly, for any vote of the manipulator his utility will be 0. So let us assume that for $x \in X$, $s - 1 \leq s(x) \leq s$. Following Obraztsova et al. [3], we define a directed graph G with a vertex set C, where there is an edge from c_i to c_j when there are exactly $s(c_j)$ voters in V' that rank c_j above c_i. We color the vertices of G as follows. Let $x \in X$ be any candidate. All the candidates $c \in C \setminus X$ with the score $s(c) = s(x) + 1$ will be purple; all the candidates $c \in C \setminus X$ with the score $s(c) = s(x)$ will be red; and all the rest of the candidates will be green. Note that by construction, all the candidates in X are green.

In order to find an optimal vote of the manipulator, we will use the recursive procedure $\mathcal{A}(H)$ described in [3], where H is an input colored directed graph, with one small modification: in step 2, if H contains any of the vertices of X, we add them (in some arbitrary order) to the top of the list L built by the procedure and remove them from H. We call this modified procedure $\mathcal{A}'(H)$.

In our algorithm, we first call $\mathcal{A}'(G)$. This way, if the expected utility of an optimal vote is greater than 0, we get an ordering L in which the number of candidates with utility 0 having the highest scores is minimal (the proof of this is the same as the original proof). Also, L contains all the candidates of X in the $|X|$ top positions. In the next step, we go over all the $|X|! \leq k!$ permutations of the candidates in X and check in which permutation the number of candidates of X whose score grows by 1 is maximal. Then we return this permutation combined with L. Note that the permutation of the candidates in X does not affect the scores of the other candidates (what really matters here is that all the candidates in X are ranked above all the other candidates). So by changing the permutation from what was calculated by $\mathcal{A}'$, we do not hurt the optimality of the solution computed by $\mathcal{A}'$.

One can verify that the running time of this algorithm is $O(k!k^2 + (n+m)m^2)$, and so, the problem is in FPT. □

Corollary 2. *When the number of 1's in the manipulator's utility vector, $k = O(\frac{\log m}{\log \log m} + \frac{\log n}{\log \log n})$ then the algorithm for finding an optimal manipulation runs in polynomial time.*

Proof. When $k = O(\frac{\log m}{\log \log m} + \frac{\log n}{\log \log n})$, $k! = O(m+n)$, and the result follows. □

3.2 Characterization Result

Here we state that the graph G of the election as defined above has some special property, which sometimes may help in computing the maximum expected utility of the manipulator.

Theorem 3. *Suppose that the utilities and the set of candidates X are as defined above. Let G be the graph of the election as defined above, and let $H = (X, E)$ be the subgraph of G induced by the vertices of X. If there exist two vertices $x, y \in X$ such that $(x, y) \in E$ and $(y, x) \in E$ then H is complete, i.e., for all $a, b \in X$, $(a, b) \in E$ and $(b, a) \in E$.*

Corollary 4. *If the conditions of Theorem 3 hold, we can compute an optimal vote of the manipulator in polynomial time.*

4. ACKNOWLEDGMENTS

We would like to thank Edith Elkind and Nati Linial for helpful discussions on the topic of this paper. This work was partially supported by Israel Ministry of Science and Technology grant #3-6797, and the Google Inter-University Center for Electronic Markets and Auctions.

5. REFERENCES

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

[2] S. Obraztsova and E. Elkind. On the complexity of voting manipulation under randomized tie-breaking. In *The Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI 2011)*, pages 319–324, 2011.

[3] S. Obraztsova, E. Elkind, and N. Hazon. Ties matter: Complexity of voting manipulation revisited. In *The Tenth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS 2011)*, pages 71–78, 2011.

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