Control of Condorcet Voting: Complexity and a Relation-Algebraic Approach

(Extended Abstract)

Rudolf Berghammer
Institut für Informatik, Universität Kiel
24118 Kiel, Germany
rub@informatik.uni-kiel.de

Henning Schnoor
Institut für Informatik, Universität Kiel
24118 Kiel, Germany
henning.schnoor@email.uni-kiel.de

ABSTRACT

We study the constructive variant of the control problem for Condorcet voting, where control is done by deleting voters. We prove that this problem remains NP-hard for the Condorcet-consistent voting rule *Uncovered Alternatives*. Furthermore, we develop a relation-algebraic model of Condorcet voting and relation-algebraic specifications of the dominance relation and the solutions of the control problem.

1. INTRODUCTION

We study *constructive control by deleting voters* [2] in an election. Here, the authority conducting the election (called the chair) knows the preferences of all voters and wants to ensure that a certain candidate a^* wins the election. To do this, she can remove voters from the election. This yields the following computational problem: Given an election consisting of a set of candidates, voters along with information on how they will vote, and a preferred candidate a^*, determine a smallest set Y of voters such that removing all voters in Y makes a^* win the election.

Following [1], numerous papers have studied the complexity of algorithmic problems in the context of elections (see e.g., [8]). For many voting rules, the control problem is NP-complete; such voting rules are deemed to be "secure" against this attempt to influence the outcome of the election. However, efficient algorithms that work for *many* cases can exist for NP-hard problems, the very successful history of SAT solvers being an impressive example (see also [7] for an example from Computational Social Choice).

We contribute to the investigation of the complexity of the control problem. We study Condorcet-consistent rules, i.e., rules which always elect a *Condorcet winner* if there is one. A Condorcet winner is a candidate a that "beats" every other candidate b in the sense that more voters prefer a to b than vice versa. We focus on the *Uncovered candidates rule* (cf. [6]). This rule states that each *uncovered* candidate wins the election in the absence of a Condorcet winner. An candidate a is uncovered if there is no candidate b such that b beats a in a pairwise contest, and every candidate c which beats b also beats a. We show that the control problem is NP-complete for this voting rule (NP-completeness of the control problem for the Condorcet voting rule was established in [2]).

To investigate whether hardness in fact precludes practical solutions to the control problem, we use the relation-algebraic tool RelView [3] to solve the control problem for both voting rules mentioned above. Since RelView represents relations as OBDDs (a special kind of branching programs), the worst-case exponential running time can be improved for practical instances. To the best of our knowledge, this is the first paper that uses relation algebra for Computational Social Choice by directly taking the voter's preferences into account. The only comparable previous results that we are aware of allow to determine the winner of an election, provided that the *dominance relation* (essentially the election's tournament-representation that abstracts away the individual votes) is precomputed (see, e.g., [4]). Our algorithms perform well also on non-toy instances. The full version of this extended abstract can be found in [5]. It also studies the Condorecet voting rule which does not elect any candidate if there is no Condorcet winner.

2. CONDORCET VOTING COMPLEXITY

An election consists of a set $N = \{1, \ldots, n\}$ of voters, a non-empty, finite set C of candidates, the preferences of each voter and a voting rule. The preferences of each voter i are expressed via a linear strict order $>_i$ on C. An *instance* of a Condorcet election consists of the sets N, C, and the relations $>_i$ for all $i \in N$. Since a Condorcet winner does not always exist, a number of so-called Condorcet-consistent voting rules (rules that elect a Condorcet winner if one exists) have been studied. For one of these rules, namely *Uncovered Alternatives* (cp. introduction), we show the following (the proof can be found in the full version [5]):

Theorem 1. *The constructive control problem by deleting voters is NP-hard for* Uncovered Alternatives.

3. RELATION-ALGEBRAIC MODELING

In the full paper [5] we present algorithms for the control problem based on relation algebra and the tool RelView, in this extended abstract we only give a brief introduction.

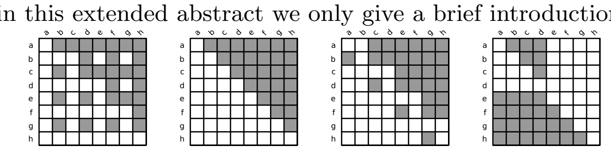

Figure 1: Individual Preferences Relations

Appears in: *Alessio Lomuscio, Paul Scerri, Ana Bazzan, and Michael Huhns (eds.), Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014), May 5-9, 2014, Paris, France.*

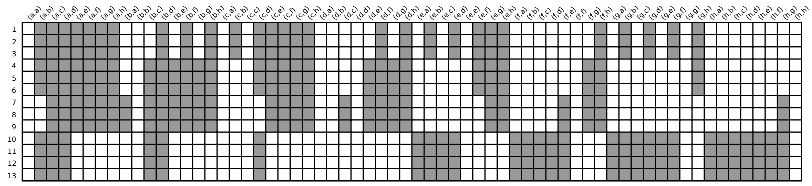

Figure 2: Election instance modeled as relation

We use binary relations, i.e., sets $R \subseteq X \times Y$ for (finite) sets X and Y, and consider such an R as a Boolean matrix. Figure 1 shows relations in matrix notation that represent 4 votes with candidate set $\{a, \ldots, h\}$. The voter with the left-most preference relation $>_1$ prefers a to all other candidates, hence the relation contains the pairs (a, x) for all $x \neq a$: there are black squares denoting that all pairs (a, x) with $x \neq a$ are elements of $>_1$. The complete preference relations specified in Figure 1 are as follows: The left-most voter ranks the candidates a, c, e, g, b, d, f, h, the second one ranks them a, b, c, d, e, f, g, h, the third one as b, a, d, c, f, e, h, g and the preference of the right-most voter is h, g, f, e, a, b, c, d.

We combine the preferences of the voters into a single relation. As an example, assume that voters 1-3 vote according to the left-most preference relation from Figure 1, voters 4-6, 7-9 and 10-13 vote according to the second, third and fourth relation from Figure 1. The combined relation can be seen in Figure 2. For each pair (x, y) of candidates, there is a column which for each voter $i \in \{1, \ldots, 13\}$ contains a positive entry (black square) if that voter prefers x to y, and a negative entry (white square) otherwise. This leads to the following definition:

Definition 3.1. *The relation* $P \subseteq N \times C^2$ models *the instance* $(N, C, (>_i)_{i \in N})$ *of an election if* $(i, (x, y)) \in P$ *if and only if* $x >_i y$*, for all* $i \in N$ *and* $(x, y) \in C^2$.

Computations in the relation-algebraic model use the basic "building blocks" of relation algebra, namely the basic relations and the operators that we will use in our computations. The basic relations are the *identity relation* I, the *universal relation* L and the *empty relation* O. More complex relations are constructed using set-theoretic operations as complement $\overline{R}$, unions $R \cup S$ and intersections $R \cap S$ and relation-algebraic operations as composition $R; S$ and transposition R^T. To give an impression what our results look like, the following theorem shows how to compute the dominance relation $D \subseteq C \times C$ of a Condorcet election. Concerning the specific relations π, ρ, M aund S and the specific relation-algebraic constructions $syq(\cdot, \cdot)$, $[\cdot, \cdot]\!]$ and $rel(\cdot)$ used in the theorem, we have again refer to the full paper [5].

Theorem 2. *Suppose that* P *models an instance of a Condorcet election. If we define relations* $E := syq(P, \mathsf{M})$*,* $F := syq(P; [\rho, \pi]\!], \mathsf{M})$ *and* $D := rel((E \cap F; (\mathsf{S} \cap \overline{\mathsf{S}^\mathsf{T}})); \mathsf{L})$*, then for all* $x, y \in C$ *we get* $(x, y) \in D$ *iff*

$$|\{i \in N \mid (i, (x, y)) \in P\}| > |\{i \in N \mid (i, (y, x)) \in P\}|.$$

Relation-algebraic specifications using the above-mentioned basic building blocks are very formal, and can be specified directly in RelView's input language. Hence we obtain our algorithms as relation-algebraic specifications which directly gives us executable code.

4. CONCLUSION

We proved a new complexity result concerning Condorcet voting, viz. that the constructive control problem by deleting voters remains NP-hard if the classical winning condition "to be a Condorcet winner" is replaced by "to be in the uncovered set." We also demonstrated how to use relation algebra for modeling and problem-solving in the area of voting rules. In experiments, our algorithms could deal with instances of "realistic" size. Our algorithms are formalized in such a way that, in principle, their automatic verification is possible.

Our results show that Computer Algebra tools can be used to obtain practical algorithms for hard problems without relying on domain knowledge for optimizations, if the data structures used in the Computer Algebra package – like the OBDDs used in RelView – automatically exploit the 'easyness' that may be present in practical instances. This supports the point of view that proving NP-hardness is not sufficient in order to conclude that a voting rule is "secure" from attempts to influence the outcome of an election. RelView also generates visualizations and has further features that support scientific experiments, like step-wise execution, test of properties and generation of random relations. All this makes the approach especially appropriate for prototyping and experimentation, and as such very instructive scientific research as well as for university education.

5. REFERENCES

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

[2] J. Bartholdi, III, C. Tovey, and M. Trick. How hard is it to control an election? *Mathematical and Computer Modeling*, 16(8/9):27–40, 1992.

[3] R. Berghammer and F. Neumann. Relview - an obdd-based computer algebra system for relations. In *CASC*, volume 3718 of *Lecture Notes in Computer Science*, pages 40–51. Springer, 2005.

[4] R. Berghammer, A. Rusinowska, and H. C. M. de Swart. Computing tournament solutions using relation algebra and relview. *European Journal of Operational Research*, 226(3):636–645, 2013.

[5] R. Berghammer and H. Schnoor. Relation-algebraic and tool-supported control of condorcet voting. *CoRR*, abs/1304.7244, 2013.

[6] F. Brandt and F. A. Fischer. Computing the minimal covering set. *Mathematical Social Sciences*, 56(2):254–268, 2008.

[7] V. Conitzer and T. Sandholm. Nonexistence of voting rules that are usually hard to manipulate. In *AAAI*, pages 627–634. AAAI Press, 2006.

[8] V. Conitzer, T. Sandholm, and J. Lang. When are elections with few candidates hard to manipulate? *J. ACM*, 54(3):14, 2007.