Discrete Applied Mathematics 162 (2014) 391–398

Contents lists available at ScienceDirect

Discrete Applied Mathematics

journal homepage: www.elsevier.com/locate/dam

Note

Binary linear programming solutions and non-approximability for control problems in voting systems

Frank Gurski *, Magnus Roos

Heinrich Heine University Düsseldorf, Institute of Computer Science, D-40225 Düsseldorf, Germany

ARTICLE INFO

Article history:
Received 8 November 2012
Received in revised form 14 August 2013
Accepted 19 August 2013
Available online 17 September 2013

Keywords:
Graph algorithms
Digraph modification problems
Binary linear programming formulations
Approximation
Computational social choice
Control problems
Voting

ABSTRACT

In this paper we consider constructive control by adding and deleting candidates in Copeland and Llull voting systems from a theoretical and an experimental point of view.

We show how to characterize the optimization versions of these four control problems as special digraph problems and binary linear programming formulations of linear size. Our digraph characterizations allow us to prove the hardness of approximations with absolute performance guarantee for optimal constructive control by deleting candidates in Copeland and by adding candidates in Llull voting schemes and the nonexistence of efficient approximation schemes for optimal constructive control by adding and deleting candidates in Copeland and Llull voting schemes. Our experimental study of running times using LP solvers shows that for a lot of practical instances even the hard control problems can be solved very efficiently.

1. Introduction

An *election* (C, V) consists of a finite set C of candidates and of a finite multiset V of votes. A *voting system* is a rule that determines the winners of a given election. In this paper we consider the voting systems, Copeland and Llull [13].

In order to change the outcome of an election several different approaches have been defined. We consider the model of control, where the task of the so-called chair is to make a favorite candidate win the election (constructive control) or to prevent a distinguished candidate from winning the election (destructive control) by adding or deleting a *bounded* number of candidates. For Copeland and Llull the time complexity of these control problems has been determined in [13].

In this paper we consider control problems $\mathcal{E}$-$\mathfrak{CT}$ defined by a voting system $\mathcal{E} \in \{\text{Copeland}, \text{Llull}\}$ and some of the four control types $\mathfrak{CT}$ obtained by constructive control or destructive control and adding or deleting candidates. Each of these control problems $\mathcal{E}$-$\mathfrak{CT}$ is transformed to an optimization version $\mathcal{E}$-O-$\mathfrak{CT}$ in which the task of the chair is to make a favorite candidate win the election (constructive control) or to prevent a distinguished candidate from winning the election (destructive control) by adding or deleting a *minimum* number of candidates. The time complexity of optimal constructive control by adding and deleting candidates in Copeland and Llull voting schemes is known to be hard. This motivates to give useful characterizations of the hard optimization problems as well as easy algorithms for their solutions. Both will be done using linear programming formulations, which is a very powerful tool with a history of more than 50 years [20]. Therefore in this paper we show how to transform all these four optimization control problems $\mathcal{E}$-O-$\mathfrak{CT}$ into equivalent digraph problems. The digraph problems are transformed into equivalent binary linear programs of linear size, i.e. using a linear number of variables and constraints with respect to the number of candidates.

While in [24] it has been shown that manipulation in Borda has an approximation algorithm with an absolute performance guarantee, we prove the nonexistence of approximation algorithms with absolute performance guarantees for

* Corresponding author. Tel.: +49 2118113713.
E-mail addresses: gurski-dam13@acs.uni-duesseldorf.de (F. Gurski), roos@cs.uni-duesseldorf.de (M. Roos).

0166-218X/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.dam.2013.08.032

Table 1
Results on control complexity of Copeland and Llull. R means resistance and V means vulnerability to a particular control type. The results are due to [13].

Control by	Constructive control (CC)		Destructive control (DC)	
	Deleting candidates (DC)	Adding candidates (AC)	Deleting candidates (DC)	Adding candidates (AC)
Copeland	R	R	V	V
Llull	R	R	V	V

optimal constructive control by deleting candidates in Copeland and by adding candidates in Llull voting schemes, unless P = NP. The known hardness of the standard parameterization of the corresponding decision problems is used to show the nonexistence of efficient approximation schemes. We have implemented our LP solutions using Matlab LP solvers. Our experimental results show that for instances up to 1000 candidates the control problems can be solved very efficiently. In particular optimal constructive control by adding candidates in Copeland and Llull voting can be done much faster than control by deleting candidates in Copeland and Llull voting.

2. Preliminaries

2.1. Elections and control problems

An *election* is a pair (C, V), where $C = \{c_1, \ldots, c_m\}$ is a finite set of candidates and $V = \{v_1, \ldots, v_n\}$ is a finite multiset of votes, where each voter expresses his or her preferences over the candidate set C. In this paper a vote is an ordered list, which is a permutation of all candidates without ties. A *voting system* is a rule that determines the winners of a given election. We use the unique-winner model, i.e. we define a candidate $c \in C$ as a winner of an election (C, V), if c is the only winner of the election (i.e., there are no other candidates tying for the first place).

In this paper we consider the voting systems Copeland and Llull, which are defined as follows. Let $\alpha \in [0, 1]$. In *Copeland*$^\alpha$ *voting* each voter has to specify a tie-free linear ordering of all candidates. Every pair of candidates (c_1, c_2) is compared within a head-to-head contest. Candidate c_1 wins against candidate c_2 in a head-to-head contest if it is better-positioned in more than half of the votes. In this case, the winner c_1 gets one point and the loser c_2 gets zero points. If the candidates are tied, they both get α points. The winner is the candidate with the highest score. Following the notations in [8], Copeland^0 is denoted as *Copeland* and Copeland^1 is denoted as *Llull*.

The outcome of an election can be affected in several ways such as with manipulation, bribery, or control. In *manipulation* [3,11] coalitions of voters cast their votes insincerely in order to reach their goal. In *bribery* [12,13] an external agent is allowed to change some voters' votes in order to reach his or her goal. In *control* [4,18,13,19] an external agent – usually called the chair – can change the structure of the election (for example by adding or deleting either candidates or voters) in order to change the outcome of the election. In this paper we are only considering control. The chair can have two different intentions. First, the chair's goal could be to make his or her favorite candidate win the election (*constructive control*) [4], second, the chair's goal could be to bar a distinguished candidate from winning the election (*destructive control*) [18]. For the voting systems $\mathcal{E} \in \{\text{Copeland, Llull}\}$ we consider the following four control problems.

Name $\mathcal{E}$-Constructive Control by Adding Candidates ($\mathcal{E}$-CC-AC)
Instance An election $(C \cup D, V)$, where C is a set of candidates and D is a set of spoiler candidates with $C \cap D = \emptyset$, a distinguished candidate $c \in C$, and a positive integer k.
Question Is there a subset $D' \subseteq D$ of size at most k, such that c is the unique $\mathcal{E}$ winner of the election $(C \cup D', V)$?

Control by adding candidates models candidate recruitment. For practical examples see [5,4,18,13]. $\mathcal{E}$-Destructive Control by Adding Candidates ($\mathcal{E}$-DC-AC) is defined analogously with the difference that we ask whether it is possible to keep candidate c from being the unique $\mathcal{E}$ winner of the election $(C \cup D', V)$.

Next we will define control by deleting candidates, which models actions where candidates are being forced out of race.

Name $\mathcal{E}$-Constructive Control by Deleting Candidates ($\mathcal{E}$-CC-DC)
Instance An election (C, V), a distinguished candidate $c \in C$, and a positive integer k.
Question Is there a subset $C' \subseteq C$ of size at most k, such that c is the unique $\mathcal{E}$ winner of the election $(C - C', V)$?

We can also define the destructive case $\mathcal{E}$-Destructive Control by Deleting Candidates ($\mathcal{E}$-DC-DC) analogously. Here we ask whether it is possible to keep candidate $c \notin C'$ from being the unique $\mathcal{E}$ winner of the election $(C - C', V)$.

The following notions are due to Bartholdi, Tovey, and Trick [4]. Let $\mathfrak{CT}$ be a control type. If the chair can never change a non winner to a unique winner in election $\mathcal{E}$ by exerting control of type $\mathfrak{CT}$, we say that $\mathcal{E}$ is *immune to* $\mathfrak{CT}$. If a voting system $\mathcal{E}$ is not immune to $\mathfrak{CT}$, then it is said to be *susceptible to* $\mathfrak{CT}$. If a voting system $\mathcal{E}$ is susceptible to $\mathfrak{CT}$, and the chair's task of controlling the election is NP-hard, $\mathcal{E}$ is said to be *resistant to* $\mathfrak{CT}$. If a voting system $\mathcal{E}$ is susceptible to $\mathfrak{CT}$, and the corresponding decision problem can be solved in polynomial time, the voting system is said to be *vulnerable to* $\mathfrak{CT}$. Table 1 shows the previous results on control complexity of Copeland and Llull, see [13].

In this paper we consider optimization versions of these decision problems, which look for a subset of candidates of minimum size which have to be removed or added in order to control the election. For the voting systems $\mathcal{E} \in \{\text{Copeland, Llull}\}$ we introduce the following four control optimization problems.

Name $\mathcal{E}$-Optimal Constructive Control by Adding Candidates ($\mathcal{E}$-O-CC-AC)
Instance An election $(C \cup D, V)$, where C is a set of candidates and D is a set of spoiler candidates with $C \cap D = \emptyset$, and a distinguished candidate $c \in C$.
Problem Find a subset $D' \subseteq D$ of minimum size, such that c is the unique $\mathcal{E}$ winner of the election $(C \cup D', V)$ or else indicate that it is impossible to do so.

$\mathcal{E}$-Optimal Destructive Control by Adding Candidates ($\mathcal{E}$-O-DC-AC) is defined analogously with the difference that we have to find a subset $D' \subseteq D$ of minimum size that keeps c from being the unique $\mathcal{E}$ winner of the election $(C \cup D', V)$.

We also define the optimization versions of control by deleting candidates.

Name $\mathcal{E}$-Optimal Constructive Control by Deleting Candidates ($\mathcal{E}$-O-CC-DC)
Instance An election (C, V) and a distinguished candidate $c \in C$.
Problem Find a subset $C' \subseteq C$ of minimum size, such that c is the unique $\mathcal{E}$ winner of the election $(C - C', V)$ or else indicate that it is impossible to do so.

We can also define the destructive case $\mathcal{E}$-Optimal Destructive Control by Deleting Candidates ($\mathcal{E}$-O-DC-DC) analogously. Here we have to find a subset $C' \subseteq C$ of minimum size that keeps $c \notin C'$ from being the unique $\mathcal{E}$ winner of the election $(C - C', V)$.

The notions *immune to* $\mathfrak{CT}$, *susceptible to* $\mathfrak{CT}$, *resistant to* $\mathfrak{CT}$ and *vulnerable to* $\mathfrak{CT}$ for some control type $\mathfrak{CT}$ can easily be extended for the optimization versions of control problems.[1] The control complexity for control optimization problems is known to be the same as for the decision problems shown in [13]. In this paper we consider the four hard optimization problems of constructive control by adding and deleting candidates in Copeland and Llull voting systems.

2.2. Elections and digraphs

A *directed graph* or *digraph* D consists of a non-empty finite set W of elements called *vertices* and a finite set A of ordered pairs of distinct vertices called *arcs*. We often briefly write $D = (W, A)$. For a vertex $v \in W$, the sets $N^+(v) = \{u \in W \mid (v, u) \in A\}$ and $N^-(v) = \{u \in W \mid (u, v) \in A\}$ are called the *set of all out-neighbors* and the *set of all in-neighbors* of v. The *outdegree* of v is the number of out-neighbors of v and the *indegree* of v is the number of in-neighbors of v. Vertex $v \in W$ has *maximum outdegree*, if there is no vertex in $W - \{v\}$ which has outdegree greater than the outdegree of v. The notion of *minimum indegree* is defined analogically. A digraph $D' = (W', A')$ is a *subdigraph* of digraph $D = (W, A)$ if $W' \subseteq W$ and $A' \subseteq A$. If every arc of A with both end vertices in W' is in A', we say that D' is an *induced subdigraph* of digraph D. We write $D' = D[W']$. A digraph $D = (W, A)$ which contains for every two vertices $x, y \in W$ at most one of the two arcs (x, y), (y, x) in A is called an *oriented graph* [2]. Obviously in an oriented graph $D = (W, A)$ there is at most one vertex of outdegree $|W| - 1$.

An election (C, V) defined by voting rules where the winners depend only on the results of head-to-head contests between voters can be encoded into an oriented graph $D = (W, A)$ as follows. The vertex set which corresponds to the candidate set C is $W = \{w_i \mid c_i \in C\}$. There is an arc from vertex w_i to w_j in A if and only if the corresponding candidate c_i defeats candidate c_j in the head-to-head contest. Thus, the Copeland score of a candidate $c_i \in C$ equals the outdegree of the corresponding vertex w_i and c_i is a Copeland winner if and only if the corresponding vertex w_i has maximum outdegree in D. The Llull score of a candidate $c_i \in C$ equals the number of all vertices $|W|$ minus the indegree of the corresponding vertex w_i minus one and thus c_i is a Llull winner if and only if the corresponding vertex w_i has minimum indegree in D.

2.3. Linear programming

Linear programming is a powerful tool, studied for over 50 years, that can be used to define a lot of very important optimization problems [10,20]. In a *linear programming problem (LP)* we are given a linear function $f : \mathbb{R}^n \mapsto \mathbb{R}, f(x_1, \ldots, x_n) = c_1x_1 + \cdots + c_nx_n = \sum_{i=1}^{n} c_ix_i$. Function f is denoted as *objective function* of the LP. Additionally, a set of constraints is given. In general a constraint either is an equality or an inequality which contains a linear combination of the variables of f, i.e. the i-th constraint is of the form $a_{i,1}x_1 + \cdots + a_{i,n}x_n \geq b_i$. Some of these constraints may be very simple, since they require that some of the variables are not negative, e.g., $x_j \geq 0$. Constraints of the first type are denoted as *functional constraints* and the latter ones are denoted as *nonnegativity constraints*.

A proposal $x'_1, \ldots, x'_n$ of values for the n variables of f is called a *solution* and $f(x'_1, \ldots, x'_n)$ its *objective (function) value*. A solution is denoted as *feasible* if it satisfies all constraints. A feasible solution x is denoted as an *optimal solution* if the objective function value for x is the smallest (minimization problem) or the largest value (maximization problem) over the objective function values for all feasible solutions. The linear programming problem is, given an objective function and a finite set of constraints, to find an optimal solution. Using matrices, a linear program can be expressed as a task of minimizing c^Tx subject to the constraints $Ax \geq b$ and $x \geq 0$. This allows us to define the *size* of a linear program by $\text{size}(A) + \text{size}(b) + \text{size}(c)$. The size of an integer (by multiplying the rational coefficients adequate we can assume that we have given integer coefficients) is the size of its binary representation. The size of a vector or of a matrix is the sum of the sizes of its elements.

[1] In [18] the notion of *(computationally) certifiably-vulnerable* is introduced for systems in which the chair can execute control in the best way, i.e. by deleting or adding the smallest number of candidates.

We define a linear program P to be *equivalent* to some optimization problem O, if for every feasible solution W_1 for P with objective value M there is a corresponding solution W_2 for O with objective value M and for every solution W_2 for O with objective value M there is a corresponding feasible solution W_1 for P with objective value M.

In *integer linear programming problems (IPs)*, the variables are all required to be integers and in *binary linear programming problems (BIPs)*, each variable can only take the two values, zero or one. While general linear programming can be solved in polynomial time by interior point methods or the ellipsoid method [10], solving integer programming and even binary integer programming is NP-hard [15].

An IP formulation of a given problem has many advantages. First, it may give a useful equivalent version of the problem (see Section 3). Further, by replacing the given constraints by weaker constraints (called a relaxation), one can compute bounds for the value of an optimal solution of the original problem or even get the possibility of designing approximation algorithms [1]. Furthermore, an IP formulation often leads to a better understanding of the relationship between different problems. Finally, IP formulations are algorithms for the given problems which can be realized by LP solvers (see Section 5).

3. Digraph characterizations and linear programming formulations

We consider constructive control by adding and deleting candidates in Copeland and Llull voting schemes. We show how to formulate the optimization versions of these four control problems as special digraph problems[2] and binary linear programming formulations. Within our programs, we will use a variable for each candidate and at-most-twice-the-number-of-candidates many constraints.

3.1. Copeland-O-CC-DC

Using our digraph model for Copeland elections, we can characterize the control problem Copeland-O-CC-DC as follows.

Name Min Max-Outdegree-Deletion (Min MOD)
Instance An oriented graph $D = (W, A)$ and a distinguished vertex $w \in W$.
Problem Find a subset $W' \subseteq W - \{w\}$ of minimum size, such that w is the only vertex in $D[W - W']$ with maximum outdegree.

Let $I = (D, w)$ (with $D = (W, A)$ and $w \in W$) be an instance of Min MOD. To formulate Min MOD as a BIP, we introduce a Boolean variable $x_v \in \{0, 1\}$ for each vertex $v \in W$. The idea is to have $x_v = 1$ if and only if v is taken into subset W'. Subset $W' \subseteq W$ is a feasible solution for Min MOD instance I if in the subgraph $D[W - W']$ the outdegree of every vertex v is smaller than outdegree of w. This can be ensured by the constraint

$$\sum_{(v,v')\in A} (1 - x_{v'}) < \sum_{(w,v')\in A} (1 - x_{v'})$$

for every vertex $v \in W - W' - \{w\}$. Since W' is unknown in the given domain of v, this constraint is not useful within a linear program. Thus we add W' to the domain of v and subtract $x_v \cdot |W|$ on the left side of the inequality.

$$\sum_{(v,v')\in A} (1 - x_{v'}) - x_v \cdot |W| < \sum_{(w,v')\in A} (1 - x_{v'})$$

for every vertex $v \in W - \{w\}$. This modified condition is obviously fulfilled for every vertex v with $x_v = 1$, i.e. for every $v \in W'$, and is equivalent to the previous inequality for every vertex v with $x_v = 0$, i.e. for every $v \notin W'$. This idea allows us to express Min MOD by the following BIP.

$$\text{minimize} \sum_{v\in W} x_v$$

subject to $x_w = 0$, $x_v \in \{0, 1\}$ for every $v \in W$, and

$$\sum_{(v,v')\in A} (1 - x_{v'}) - x_v \cdot |W| < \sum_{(w,v')\in A} (1 - x_{v'})$$

for every $v \in W - \{w\}$.

3.2. Copeland-O-CC-AC

Next we consider Copeland-O-CC-AC which can be characterized by the following digraph modification problem.

Name Min Max-Outdegree-Addition (Min MOA)
Instance An oriented graph $D = (W, A)$ with vertex set $W = C \cup N$ and a distinguished vertex $c \in C$.
Problem Find a subset $N' \subseteq N$ of minimum size, such that c is the only vertex in $D[C \cup N']$ with maximum outdegree.

[2] A similar construction was given in [8] for the corresponding decision problems.

Let $I = (D, c)$ with $D = (W, A)$, $W = C \cup N$ and $c \in C$ be an instance of Min MOA. We use a Boolean variable $x_v \in \{0, 1\}$ for each vertex $v \in W$. The idea is to have $x_v = 1$ if and only if v is in C or taken into subset N'.[3] This idea allows us to define Min MOA by minimizing $\sum_{v \in W} x_v$ subject to $x_v = 1$ for every $v \in C$, $x_v \in \{0, 1\}$ for every $v \in W$, and $\sum_{(v,v') \in A} x_{v'} - (1 - x_v) \cdot |W| < \sum_{(c,v') \in A} x_{v'}$ for every $v \in W - \{c\}$.

3.3. Llull-O-CC-DC

In order to define control in Llull in our digraph model we have to consider the indegree of the vertices. Thus Llull-O-CC-DC can be characterized by the following digraph problem.

Name Min Min-Indegree-Deletion (Min MID)
Instance An oriented graph $D = (W, A)$ and a distinguished vertex $w \in W$.
Problem Find a subset $W' \subseteq W - \{w\}$ of minimum size, such that w is the only vertex in $D[W - W']$ with minimum indegree.

Let $I = (D, w)$ with $D = (W, A)$ and $w \in W$ be an instance of Min MID. We define a Boolean variable $x_v \in \{0, 1\}$ for every vertex $v \in W$. The idea is to have $x_v = 1$ if and only if v is taken into subset W'. Thus Min MID corresponds to minimizing $\sum_{v \in W} x_v$ subject to $x_w = 0$, $x_v \in \{0, 1\}$ for every $v \in W$, and $\sum_{(v',v) \in A}(1 - x_{v'}) + x_v \cdot |W| > \sum_{(v',w) \in A}(1 - x_{v'})$ for every $v \in W - \{w\}$.

Further results on the interesting graph problem of how to make a distinguished vertex have minimum indegree by deleting vertices can be found in [7].

3.4. Llull-O-CC-AC

Problem Llull-O-CC-AC can be characterized by the following digraph modification problem.

Name Min Min-Indegree-Addition (Min MIA)
Instance An oriented graph $D = (W, A)$ with vertex set $W = C \cup N$ and a distinguished vertex $c \in C$.
Problem Find a subset $N' \subseteq N$ of minimum size, such that c is the only vertex in $D[C \cup N']$ with minimum indegree.

Let $I = (D, c)$ with $D = (W, A)$, $W = C \cup N$ and $c \in C$ be an instance of Min MIA. We define a Boolean variable $x_v \in \{0, 1\}$ for every vertex $v \in W$. The idea is to have $x_v = 1$ if and only if v is in C or taken into subset N'. Min MIA corresponds to minimizing $\sum_{v \in W} x_v$ subject to $x_v = 1$ for every $v \in C$, $x_v \in \{0, 1\}$ for every $v \in W$, and $\sum_{(v',v) \in A} x_{v'} + (1 - x_v) \cdot |W| > \sum_{(v',c) \in A} x_{v'}$ for every $v \in W - \{c\}$.

4. Approximation results

Let Π be some optimization problem and I be some instance of Π. By $\mathrm{OPT}(I)$ we denote the value of an optimal solution for Π on input I. An *approximation algorithm* A for Π is an algorithm which returns in polynomial time a feasible solution for Π. The value of the solution of A on input I is denoted by $A(I)$.

We next discuss special kinds of approximations for our digraph problems.

4.1. Hardness of absolute approximation

Approximation algorithm A has *absolute performance guarantee* if there exists some positive integer k such that for every instance I of Π it holds that $|A(I) - \mathrm{OPT}(I)| \leq k$. Examples of problems which allow algorithms with absolute performance guarantee are vertex- and edge-coloring problems [1].

Next we show by a gap amplification the hardness of absolute approximation for Min MOD.

Theorem 4.1. *There is no approximation algorithm with absolute performance guarantee for* Min MOD, *unless* P = NP.

Proof. Let $I = (D, w)$, $D = (W, A)$ be an instance of Min MOD. For every positive integer k we define a further instance $I^k = (D^k, w^1)$ as follows. Oriented graph D^k is the disjoint union of first one copy of D and then $k - 1$ copies of $D[W - \{w\}]$. For $u \in W$ and $1 \leq i \leq k$ we denote by u^i the corresponding vertex in copy number i. For every $v \in N^-(w)$ for $1 < i \leq k$ we insert an arc (v^i, w^1) into D^k. The distinguished vertex of I^k is vertex w of the first copy, i.e. w^1. By this construction, the outdegree of every vertex in oriented graph D^k is the same as for its corresponding vertex in D. Although the indegree of w may increase and the indegree of the out-neighbors of w in the $k - 1$ copies of $D[W - \{w\}]$ may decrease, for every positive

[3] Note that within our BIP for control problems defined by adding candidates, vertices v with $x_v = 1$ are chosen in the considered subgraph, while in BIP for control problems defined by deleting candidates, vertices v with $x_v = 1$ are removed from the graph.

integer k it holds that $\mathrm{OPT}(I) = s \Leftrightarrow \mathrm{OPT}(I^k) = k \cdot s$. The proof of such an equivalence is straightforward by the definition of I^k, see [1] for similar proofs.

Let us assume that there is an approximation algorithm with performance guarantee k for Min MOD. Given some instance I of Min MOD we apply A on instance I^{k+1}. Then the following inequalities hold true.

$$A(I^{k+1}) - \mathrm{OPT}(I^{k+1}) \le k$$
$$A(I^{k+1}) - (k+1) \cdot \mathrm{OPT}(I) \le k$$
$$\frac{A(I^{k+1})}{k+1} - \mathrm{OPT}(I) \le \frac{k}{k+1} < 1.$$

For every feasible solution of size $A(I^{k+1})$ for instance I^{k+1} we obviously can find in polynomial time a feasible solution W' of size $\frac{A(I^{k+1})}{k+1}$ for instance I. That is, we have found a feasible solution W' for I such that $|W'| - \mathrm{OPT}(I) < 1$. Since the values $|W'|$ and $\mathrm{OPT}(I)$ are positive integers they have to be equal and we have found an optimal solution in polynomial time which implies that our assumption is incorrect. □

Theorem 4.2. *There is no approximation algorithm with absolute performance guarantee for* Min MIA, *unless* P = NP.

Proof. Let $I = (D, c)$, $D = (W, A)$, $W = C \cup N$ be an instance of Min MIA. Every optimal solution N' for I is a subset of the set of all non-in-neighbors of c, thus we can assume that $(v, c) \notin A$ for every $v \in N$.

For every positive integer k we define a further instance $I^k = (D^k, c^1)$. Oriented graph D^k is the disjoint union of first one copy of D and then $k - 1$ copies of $D[W - \{c\}]$. For every $v \in N^+(c)$ for $1 < i \le k$ we insert an arc (c^1, v^i) into D^k. The distinguished vertex of I^k is vertex c of the first copy, i.e. c^1. By our construction for every positive integer k it holds that $\mathrm{OPT}(I) = s \Leftrightarrow \mathrm{OPT}(I^k) = k \cdot s$.

The rest of the proof can be done in the same way as shown in the proof of Theorem 4.1. □

We suppose that Theorems 4.1 and 4.2 also hold for Min MID and for Min MOA but here the definition of an appropriately oriented graph D^k seems to be quite difficult. By our digraph characterizations we also have shown the following result.

Corollary 4.3. *There is no approximation algorithm with absolute performance guarantee for optimal constructive control by deleting candidates in Copeland and by adding candidates in Llull voting schemes, unless* P = NP.

In [24] it has been shown that manipulation in Borda has an approximation algorithm with absolute performance guarantee. Further results on approximations of manipulating elections can be found in [9].

4.2. Nonexistence of efficient approximation schemes

Approximation algorithm A has a *relative performance guarantee*, if there exists a positive integer k, such that for every instance I of Π it holds that $\max\{\frac{A(I)}{\mathrm{OPT}(I)}, \frac{\mathrm{OPT}(I)}{A(I)}\} \le k$. The existence of an approximation algorithm with relative performance guarantee for our four digraph problems is open until now.

A *polynomial-time approximation scheme (PTAS)* for Π is an algorithm A, for which the input consists of an instance of Π and some ϵ, $0 < \epsilon < 1$, such that for every fixed ϵ algorithm A is an approximation algorithm with relative performance guarantee $1 + \epsilon$. An *efficient polynomial-time approximation scheme (EPTAS)* is a PTAS running in time $|I|^{\mathcal{O}(1)} \cdot f(1/\epsilon)$. A *fully polynomial-time approximation scheme (FPTAS)* is a PTAS running in time $|I|^{\mathcal{O}(1)} \cdot (1/\epsilon)^{\mathcal{O}(1)}$. Obviously every FPTAS is an EPTAS and every EPTAS is a PTAS.

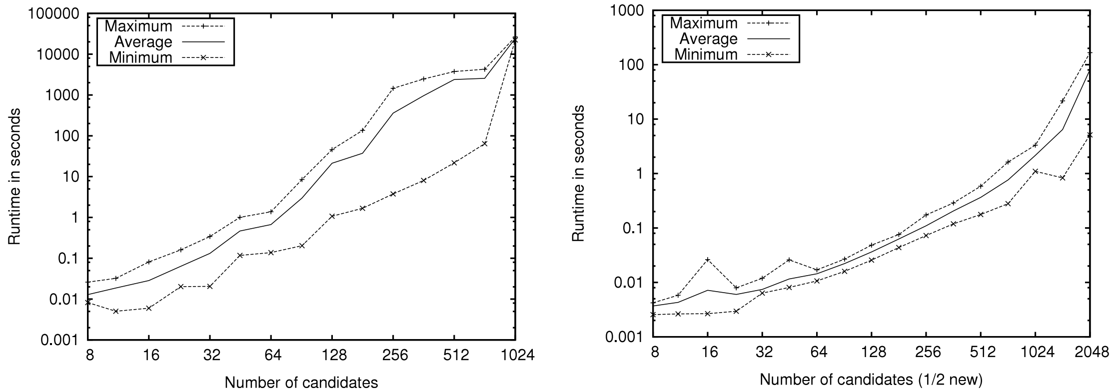
If some optimization problem Π has an EPTAS, then the standard parameterization of the corresponding decision problem k-Π is fixed parameter tractable (fpt) w.r.t. standard parameter k, i.e., solvable in time $|I|^{\mathcal{O}(1)} \cdot f(k)$, see Theorem 1.32 in [14]. In [8] it is shown that MOD, MID, MOA, and MIA are W[2]-complete w.r.t. standard parameter k, which implies that the corresponding optimization problems Min MOD, Min MID, Min MOA, and Min MIA do not have an EPTAS, unless the W-hierarchy collapses.

Corollary 4.4. *There is no EPTAS (and thus no FPTAS) for optimal constructive control by adding and deleting candidates in Copeland and Llull voting schemes, unless* W[2] = FPT.

5. Experimental results

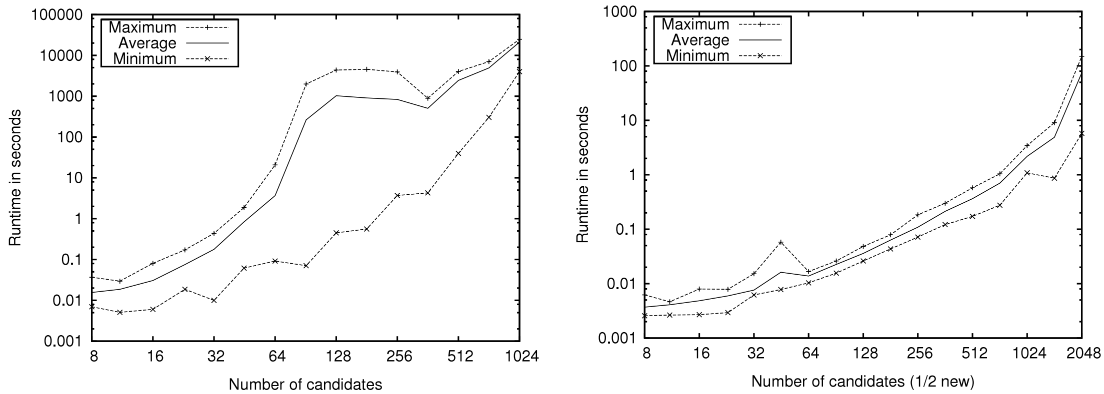
Using Matlab LP solvers on a standard desktop PC, we have compared the running time of finding optimal solutions for hard control problems and different number of candidates within Copeland and Llull. For each number of candidates m the maximum, average, and minimum curves are derived from 10 randomly generated elections, see Figs. 1 and 2.

In more detail, every such election is obtained by n randomly generated permutations (votes) of the m elements (candidates), where n was chosen to be in the interval $[m, \sqrt{m}^3]$ at random for each election. Please note that – in contrast to the generation of the integer program – the running time to solve the related integer programs does not depend on the number of voters n.



(a) Copeland-O-CC-DC. (b) Copeland-O-CC-AC.

Fig. 1. Running times for finding optimal solutions for control problems in Copeland elections.



(a) Llull-O-CC-DC. (b) Llull-O-CC-AC.

Fig. 2. Running times for finding optimal solutions for control problems in Llull elections.

For the problems of adding candidates the set of candidates is divided into two sets C and D, where C denotes the old candidates and D denotes the new ones. For the simulation we used $\|C\| = \|D\| = m/2$.

In this paper we generated elections randomly, which is known as the Impartial Culture model [17]. There are other approaches of generating random elections, like the Impartial Anonymous Culture model [21,16] or the Polya Eggenberger urn model [6], where copies of each drawn vote are added into the urn. Since we do not investigate the outcome of the elections but only the running time, we claim that the Impartial Culture model is suitable for our purpose. We interpret the visible outliers in the curves by the limited number of 10 runs we used. But if you have a look at the figures, running times of several hours occurred. Moreover, we only measured the time to solve the NP-complete binary integer program, not the time to convert the generated election into the BIP (which also yields an $\mathcal{O}(n^2 \cdot m)$-algorithm).

To summarize, what can be seen in the figures? Remarkable is the behavior of optimal constructive control by adding candidates in Copeland and Llull votings. Although both problems are NP-hard in the worst case, they can be solved within up to two minutes for practical cases, where a maximum of $n = 1024$ candidates participate in the election and the chair is allowed to add the same amount of new candidates. This suggests that this complexity only occurs in the worst case. This behavior has been researched in many papers for manipulation of voters, see e.g. [23,22]. Regarding the cases of deleting candidates, we can see a considerably higher running time, but nevertheless no exponential blowup within the range of 1000 candidates.

6. Conclusions and outlook

We gave useful equivalent digraph problems and binary linear programming formulations for several control problems in elections. Our programs should help to give a deeper understanding in the definitions and relationships of the considered control problems. Since such programs are independent of the voting system notions they might also be interesting for researchers without any background knowledge in voting systems. Further, these formulations lead to quite simple but exponential time algorithms for computing the optimal control for practical problem sizes using LP-solvers within a few seconds.

From a theoretical point of view, there are several interesting open questions. Is it possible to find an approximation algorithm with absolute performance guarantee for Min MID and for Min MOA? For the other two hard control problems this is shown to be impossible in Section 4.1. Is it possible to design approximation algorithms with a relative performance guarantee, e.g. by a relaxation of the given linear programs? Are there any special digraph classes (e.g. directed trees) for which hard control problems become polynomial even in the worst case? Some results in this direction have been shown in [8]. What is the average time complexity of Copeland-O-CC-AC and Llull-O-CC-AC? Are there any interesting results for the graph modification problems on undirected graphs? In [7] MID was also considered for undirected graphs.

References

[1] G. Ausiello, P. Crescenzi, G. Gambosi, V. Kann, A. Marchetti-Spaccamela, M. Protasi, Complexity and Approximation: Combinatorial Optimization Problems and their Approximability Properties, Springer-Verlag, Berlin, 1999.
[2] J. Bang-Jensen, G. Gutin, Digraphs. Theory, Algorithms and Applications, Springer-Verlag, Berlin, 2009.
[3] J. Bartholdi III, C. Tovey, M. Trick, The computational difficulty of manipulating an election, Social Choice and Welfare 6 (3) (1989) 227–241.
[4] J. Bartholdi III, C. Tovey, M. Trick, How hard is it to control an election? Mathematical and Computer Modelling 16 (8/9) (1992) 27–40.
[5] D. Baumeister, G. Erdélyi, E. Hemaspaandra, L. Hemaspaandra, J. Rothe, Computational aspects of approval voting, in: J. Laslier, R. Sanver (Eds.), Handbook on Approval Voting, Springer-Verlag, 2010, pp. 199–251 (Chapter 10).
[6] S. Berg, Paradox of voting under an urn model: the effect of homogeneity, Public Choice 47 (2) (1985) 377–387.
[7] N. Betzler, R. Bredereck, R. Niedermeier, J. Uhlmann, On making a distinguished vertex minimum degree by vertex deletion, in: Proceedings of the Conference on Current Trends in Theory and Practice of Computer Science, in: LNCS, Springer-Verlag, 2011, pp. 123–134.
[8] N. Betzler, J. Uhlmann, Parameterized complexity of candidate control in elections and related digraph problems, Theoretical Computer Science 410 (2009) 5425–5442.
[9] E. Brelsford, P. Faliszewski, E. Hemaspaandra, H. Schnoor, I. Schnoor, Approximability of manipulating elections, in: Proceedings of the 23st International Joint Conference on Artificial Intelligence, IJCAI/AAAI, 2008, pp. 44–49.
[10] V. Chvátal, Linear Programming, W. H. Freeman and Company, New York, 1983.
[11] V. Conitzer, T. Sandholm, J. Lang, When are elections with few candidates hard to manipulate? Journal of the ACM 54 (3) (2007) Article 14.
[12] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, The complexity of bribery in elections, Journal of Artificial Intelligence Research 35 (2009) 485–532.
[13] P. Faliszewski, E. Hemaspaandra, L. Hemaspaandra, J. Rothe, Llull and Copeland voting computationally resist bribery and control, Journal of Artificial Intelligence Research 35 (2009) 275–341.
[14] J. Flum, M. Grohe, Parameterized Complexity Theory, Springer-Verlag, Berlin, 2006.
[15] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman and Company, San Francisco, 1979.
[16] W.V. Gehrlein, P.C. Fishburn, Condorcet's paradox and anonymous preference profiles, Public Choice 26 (1976) 1–18.
[17] G.T. Guilbaud, Les théories de l'intérêt général et le problème logique de l'agrégation, Économie Appliquée 5 (1952) 501–584.
[18] E. Hemaspaandra, L. Hemaspaandra, J. Rothe, Anyone but him: the complexity of precluding an alternative, Artificial Intelligence 171 (5–6) (2007) 255–285.
[19] E. Hemaspaandra, L. Hemaspaandra, J. Rothe, Hybrid elections broaden complexity-theoretic resistance to control, Mathematical Logic Quarterly 55 (4) (2009) 397–424.
[20] M. Jünger, T.M. Liebling, D. Naddef, G.L. Nemhauser, W.R. Pulleyblank, G. Reinelt, G. Rinaldi, L.A. Wolsey (Eds.), 50 Years of Integer Programming 1958–2008, Springer-Verlag, 2010.
[21] K. Kuga, H. Nagatani, Voter antagonism and the paradox of voting, Econometrica 42 (6) (1974) 1045–1067.
[22] T. Walsh, Where are the really hard manipulation problems? The phase transition in manipulating the veto rule, in: Proceedings of the 21st International Joint Conference on Artificial Intelligence, IJCAI/AAAI, 2009, pp. 324–329.
[23] L. Xia, V. Conitzer, A sufficient condition for voting rules to be frequently manipulable, in: ACM Conference on Electronic Commerce, ACM, 2008, pp. 99–108.
[24] M. Zuckerman, A.D. Procaccia, J.S. Rosenschein, Algorithms for the coalitional manipulation problem, in: Proceedings of the ACM–SIAM Symposium on Discrete Algorithms, ACM–SIAM, 2008, pp. 277–286.