Artificial Intelligence 259 (2018) 32–51

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint

Optimal defense against election control by deleting voter groups

Yue Yin [a,*], Yevgeniy Vorobeychik [b,**], Bo An [c], Noam Hazon [d]

[a] *Key Lab of Intelligent Information Processing, ICT, CAS, University of Chinese Academy of Sciences, Beijing, China*
[b] *Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN, United States*
[c] *School of Computer Science and Engineering, Nanyang Technological University, Singapore*
[d] *Dept. of Computer Science, Ariel University, Israel*

ARTICLE INFO

Article history:
Received 5 June 2017
Received in revised form 7 February 2018
Accepted 15 February 2018
Available online 21 February 2018

Keywords:
Election control
Protecting elections
Security games

ABSTRACT

Election control encompasses attempts from an external agent to alter the structure of an election in order to change its outcome. This problem is both a fundamental theoretical problem in social choice, and a major practical concern for democratic institutions. Consequently, this issue has received considerable attention, particularly as it pertains to different voting rules. In contrast, the problem of how election control can be prevented or deterred has been largely ignored. We introduce the problem of optimal defense against election control, including destructive and constructive control, where manipulation is allowed at the granularity of groups of voters (e.g., voting locations) through a denial-of-service attack, and the defender allocates limited protection resources to prevent control. We consider plurality voting, and show that it is computationally hard to prevent both types of control, though destructive control itself can be performed in polynomial time. For defense against destructive control, we present a double-oracle framework for computing an optimal prevention strategy. We show that both defender and attacker best response subproblems are NP-complete, and develop exact mixed-integer linear programming approaches for solving these, as well as fast heuristic methods. We then extend this general approach to develop effective algorithmic solutions for defense against constructive control. Finally, we generalize the model and algorithmic approaches to consider uncertainty about voter preferences. Experiments conducted on both synthetic and real data demonstrate that the proposed computational framework can scale to realistic problem instances.[1]

* Principle corresponding author.
** Corresponding author.
E-mail addresses: melody1235813@gmail.com (Y. Yin), yevgeniy.vorobeychik@vanderbilt.edu (Y. Vorobeychik), boan@ntu.edu.sg (B. An), noamh@ariel.ac.il (N. Hazon).

[1] A preliminary version of this work appeared in the Proceedings of the 25th International Joint Conference on Artificial Intelligence (IJCAI-16) [68]. There are several major advances in this article: (i) Whereas the earlier work considered only defense against destructive control, we extended the model to consider defense against constructive control as well. (ii) We analyzed the time complexity of preventing constructive control, and proposed a heuristic algorithm which computes the defense strategy to prevent constructive control in polynomial time (under certain circumstances). (iii) We proposed a mixed integer linear programming approach to compute a strong Stackelberg equilibrium of the game of defense against constructive control, and a more scalable approximation algorithm. (iv) We performed detailed experiments to evaluate the proposed algorithms.

https://doi.org/10.1016/j.artint.2018.02.001
0004-3702/© 2018 Elsevier B.V. All rights reserved.

1. Introduction

Democratic institutions rely on the integrity of the voting process. A major threat to this integrity is the possibility that the process can be subverted by malicious parties for their own ends. Indeed, actual incidents of demonstrated and real attempts at vote manipulation and control bear out this concern. For example, the 2013 election in Pakistan was marred by a series of election-day bombings and shootings, resulting in over 150 people dead or injured, in an attempt to subvert the voting process [57], and the 2010 Sri Lanka election exhibited 84 major and 202 minor incidents of poll-related violence [7]. With the dawn of electronic and Internet voting, the additional threat of election control and manipulation through cyber attacks on electronic and Internet voting systems has emerged, with a number of documented demonstration attacks [3, 62]. While recent allegations in the U.S. of deliberate cyber attacks aimed at influencing election outcomes have not been specifically against electronic voting systems [61], the collective evidence suggests that these are vulnerable, and may well become targets in the near future. Consequently, cyber security experts have repeatedly urged deployment of better auditing systems for electronic voting systems used in the U.S. elections, arguing that elections can be successfully manipulated through targeted attacks at voting machines and polling places, for example, in battleground states [56]. However, auditing all such systems is laborious and expensive, and few systematic methods for selective auditing have been deployed to date [33].

We consider the general problem of protecting elections against malicious subversion, for example, through deployment of enhanced physical security at voting locations, or strategies for choosing which electronic voting machines or voting districts to audit. We model attacks on elections as targeting *voter groups*, such as voting machines or precincts, in order to change the identity of the election winner, a problem commonly known in prior literature as *election control*.

The study of the computational complexity of election control was initiated by Bartholdi et al. [4], who considered the problem from the perspective of computational complexity. Since then it has received considerable attention in prior literature (see Section 2). In this literature, a voting rule is viewed as resistant if control is NP-hard, and vulnerable otherwise. Many voting rules were shown to be resistant to several types of control, while plurality—which is widely used—can be controlled through voter deletion in polynomial time [4,27]. However, control is usually studied at the granularity of individual voters, and protection, when considered, is about designing voting rules which are NP-hard to control [19,28]. While these considerations are crucial if one is to understand vulnerability of elections, they are also limited in several respects. First, as the incidents of control described above attest, control can be exercised for groups of voters through a single attack, such as a denial-of-service attack on a voting station or a polling center (of which bombing is an extreme example). Second, NP-hardness of control is insufficient evidence for resistance: it is often possible to solve large instances of NP-hard problems in practice (see, e.g., [64] in the case of SAT). Resistance to election control in the broader sense, such as through allocation of limited protection resources to prevent attacks on specific voter groups, has, to our knowledge, neither been modeled nor investigated to date.

To address these limitations, we consider the problem of optimally protecting elections against control. We model control as a denial-of-service (deletion) attack on a subset of voter *groups*, which may represent polling places or electronic voting stations, with the goal of preventing the original winner from winning (*destructive* control) or making another candidate win (*constructive* control). We focus on plurality voting. Erdélyi et al. [18] show that constructive control by adding, deleting, and partitioning voters in this model is NP-hard, and recently Maushagen and Rothe [45] have shown NP-completeness of constructive and destructive control by partitioning voter groups for each non-trivial pure scoring rule. In contrast, we show that destructive control can be decided in polynomial time. We then consider the problem of defense against both types of control, modeling it as a Stackelberg game in which an outside party deploys limited protection resources to protect a collection of voter groups, allowing for randomization, and the adversary responds by attempting to subvert (control) the election. Specifically, defense against destructive control is modeled as a zero-sum game, while defense against constructive control is modeled as a nonzero-sum game. Protection resources may represent actual physical security for polling centers or voting stations, or resources devoted to auditing of specific electronic voting systems or electoral districts. We assume that the defender's goal is to ensure that the same candidate wins with or without an election control attack. We show that the problem of choosing the minimal set of resources to guarantee that an election cannot be controlled is computationally hard for both destructive and constructive control. For general cases of defense against destructive control, we propose a double-oracle framework to compute an optimal protection. We prove that both the defender and attacker oracles are NP-complete when randomized strategies are allowed. On the positive side, we develop novel mixed-integer linear programming formulations for both oracles that enable us to compute a provably optimal solution for protecting elections from destructive control. Moreover, we develop heuristic defender and attacker oracles which significantly speed up computation in the framework. Our experiments demonstrate the effectiveness and scalability of our algorithmic approach.

For defense against constructive control, we first introduce a heuristic algorithm which can compute the optimal defender strategies in polynomial time for certain cases. We then show that though the problem is modeled as a nonzero-sum game, we can achieve an approximately optimal solution by solving two zero-sum games. Indeed, experiments show that this approach computes an *optimal* solution to the original non-zero-sum game in nearly all cases.

In summary, we make the following contributions:

- A new model for protecting elections from group-level election control attacks.
- A polynomial-time algorithm for group-level destructive control.

- Complexity analysis of guaranteeing that an election cannot be controlled, including destructive and constructive control.
- A scalable double-oracle framework for choosing an optimal allocation of protection resources to prevent destructive control.
- Heuristic and approximate algorithms for computing defender strategies to prevent constructive control.

2. Related work

The study of the computational complexity of election control was initiated by Bartholdi et al. [4], who analyzed plurality and Condorcet voting with several types of control. While Bartholdi et al. studied only the constructive variant of the control problem, where the goal is to ensure a given candidate's victory, we also study the destructive variant of control, where the goal is to prevent the current winner from winning. The destructive variant of control was introduced by Hemaspaandra et al. [27], who also analyzed the approval voting rule. The study of election control was further extended to a number of other models and voting rules [6,39,40,21,51,22,42,30,29,50,67,65,45,20,32,49,44,41,15,16,53,8,66,31,17,43,25,24,23,9,55]. However, all of these consider the election control problem at the granularity of individual voters. Menton [48] considered range voting in which a voter can assign certain scores to each candidate. Consequently, a voter in this model is similar to a "group" in our setting. Unlike our model, this work imposes an external upper bound (range) on the score that a voter can assign to any candidate.

Bulteau et al. [10] were the first to consider group-level election control, which they termed combinatorial voter control. They consider control when bundles of voters need to be added and the voters are grouped according to a given bundling function, which allows for overlapping groups. In our setting the voters are grouped with no overlap, and the election control is by deleting groups of voters. Recently, Erdélyi et al. [18] studied election control of plurality by adding or deleting groups of voters, but they only consider constructive control. Finally, Chen et al. [11] studied constructive and destructive control by adding or deleting groups of *candidates* (but not voters). We note that from a broader perspective, the control in voting is similar to notion of control used in other related fields as well, such as judgment aggregation [5], weighted voting games [69,54] and fair division [2].

Work by Elkind and Lipmaa [14] is closely related in spirit to our endeavor in trying to induce hardness of manipulation of voting protocols through design. However, their focus is on manipulation, rather than control, and on designing rules which are NP-hard to manipulate, rather than using limited resources to reduce the likelihood of successful subversion of an election.

There has been extensive research on modeling physical security problems using Stackelberg games [58,63,13]. Much of prior work has focused on attackers who can only attack a single target [26,1]. Exceptions to this involved either simultaneous-move scenarios [36] or heuristic approaches [59]. In contrast, we consider adversaries attacking multiple targets (by deleting subsets of voter groups), solving the problem to optimality. In addition, the payoff structure in prior work is typically tied to the assumption of single-target attacks, whereas payoffs in our setting depend on whether deleted voter groups can affect voting outcomes. Double-oracle methods have been previously proposed for solving large Stackelberg security games [47,34,60]. However, as oracles are model dependent, the special structure of our problem requires novel scalable algorithms for solving oracles.

3. Background: Stackelberg security games

A generic Stackelberg game has two players, a *leader* and a $follower$. The leader plays first. The follower then plays based on the leader's action [12]. The two players in a Stackelberg game need not represent individuals, but could also be groups that cooperate to execute a joint strategy, such as a police force or a terrorist organization. Each player has a set of possible *pure strategies*, representing the actions that they can execute. Payoffs for each player are defined over all possible joint pure-strategy outcomes. A mixed strategy is a probability distribution over pure strategies. The payoff functions can be extended to mixed strategies by taking the expectation over pure-strategy outcomes. Playing a mixed strategy allows a player to randomize his actions, and in most cases increase his payoffs. By playing later, the follower has a knowledge of the leader's strategy (specifically, knowledge of the distribution over pure strategies chosen by the defender if the defender plays a mixed strategy), and can act accordingly to optimize his own payoffs.

Stackelberg games are used to model the defender-attacker strategic interaction in security domains and this class of Stackelberg games (with certain restrictions on payoffs) is called Stackelberg security games [58]. In the Stackelberg security game framework, the defender (e.g., security force) is modeled as the leader and the attacker (e.g., adversaries) is in the role of the follower. In this scenario, there is usually a set $\mathcal{I}$ of n targets. A pure strategy of the defender, defined as s, is to protect some of the targets; while a pure strategy of the attacker, defined as a, is to attack some of the targets. Given a strategy pair (s, a), the payoff of the defender can be represented as $u^D(s, a)$, while the payoff of the attacker can be represented as $u^A(s, a)$. Let $\mathcal{S}$ and $\mathcal{A}$ represent the strategy space of the defender and the attacker respectively. Each player has a payoff matrix based on each other's strategies.

A mixed strategy of the defender is defined as $\mathbf{x} = \{x_s : s \in \mathcal{S}\}$, with x_s representing the probability that pure strategy s is played. For the follower in the Stackelberg game, the optimal response to the defender's action is always a pure strategy, thus only pure strategies of the attacker are considered. Given a strategy pair $(\mathbf{x}, a)$, the payoff of the defender can be defined as $u^D(\mathbf{x}, a) = \sum_{s \in \mathcal{S}} x_s U^D(s, a)$; while the payoff of the attacker can be defined as $u^A(\mathbf{x}, a) = \sum_{s \in \mathcal{S}} x_s U^A(\mathbf{x}, a)$.

The Strong Stackelberg Equilibrium (SSE) is usually adopted as the solution concept in Stackelberg security games [1]. In this concept, the attacker responds the best to the defender's strategy, while breaking ties in favor of the defender.[2] The defender chooses a strategy to optimize her utility given the attacker's best response. Formally, the Strong Stackelberg Equilibrium is defined as follows.

Definition 1. Let $g(\mathbf{x}) : \mathbf{x} \to a$ be the attacker's response function. A strategy pair $(\mathbf{x}^*, g(\mathbf{x}^*))$ forms SSE if it satisfies that

1. The defender plays a best response: $u^D(\mathbf{x}^*, g(\mathbf{x}^*)) \geq u^D(\mathbf{x}, g(\mathbf{x}))$ for any other $\mathbf{x}$.
2. The attacker plays a best response: $g(\mathbf{x}^*) \in f(\mathbf{x}^*)$ where $f(\mathbf{x}^*) = \text{argmax}_{a \in \mathcal{A}} u^A(\mathbf{x}^*, a)$ is the set of attacker's best responses.
3. The attacker breaks ties in favor of the defender: $u^D(\mathbf{x}^*, g(\mathbf{x}^*)) \geq u^D(\mathbf{x}^*, a)$, $\forall a \in f(\mathbf{x}^*)$.

Many in the literature also use an extension to the generic Stackelberg games, i.e., Bayesian Stackelberg games, to model the defender-attacker strategic interaction [35,52]. Such games allow multiple 'types' of attackers. Each type of attacker has his own strategy space, hence payoff matrix. A type has a prior probability representing the likelihood of its occurrence. The defender commits to a mixed strategy, knowing only the prior distribution but not the exact type of the attacker she faces. The Strong Stackelberg Equilibrium in Bayesian Stackelberg games is similar with Definition 1, where the leader maximizes her expected utility over all types of attackers, assuming the attacker of each type chooses the best response and breaks ties in favor of the defender.

4. Model

Our model of attacks on elections is based on the prior literature investigating *election control*. In our model, control occurs at the granularity of voter groups. To begin, we assume that both the attacker and defender know the net voting tallies (but not necessarily individual votes) for each voter group. We relax this assumption in Section 7. Formally, suppose that there is a set I of n non-overlapping groups of voters and a set of candidates C over which voters have preferences. For each group $i \in I$, let t_{ic} be the tally of votes for candidate c in this group. Let $t_c = \sum_i t_{ic}$ be the total vote tally for $c \in C$. Let $\omega \in C$ be the candidate who would have won with the original set of voters: $\omega = \arg\max_c t_c$. The *attacker* can *delete* a subset of at most k voter groups to disenfranchise voters in these groups (for example, attacking k polling stations).

We study both *destructive control*, in which the attacker's goal is to prevent the original winner from winning the election, and *constructive control*, in which the attacker wants to let a specific candidate other than the original winner win the election. Our investigation is focused on *plurality voting*, in which only a single candidate is selected by each voter, and the candidate with the most votes wins (we assume that the tie-breaking rule is adversarial to the defender). Once a group is attacked and deleted, all votes from members of this group are removed, and the total tally achieved by each candidate changes correspondingly. In the case of *destructive control*, the attacker's goal is to prevent ω from winning. In *constructive control*, the attacker's goal is to make a candidate of their choice $\upsilon \in C$ $(\upsilon \neq \omega)$ win. The defender's goal is to defend against each type of control, i.e., to ensure that the original winner still wins the election despite the control. Formally, the problems are defined as follows.

Definition 2 *(Group-level destructive control).* Given a set C of candidates, a set V of voters represented by their preference order, a set I of non-overlapping groups of voters, and a non-negative integer k. Let ω be the candidate that wins under plurality rule when all the voters of V vote. We are asked whether there exists a set of voters groups $S_a \subseteq I$, $|S_a| \leq k$, such that if we remove from V all the voters in S_a then the plurality winner is not ω.

Definition 3 *(Group-level constructive control).* Given a set C of candidates, a set V of voters represented by their preference order, a set I of non-overlapping groups of voters, and a non-negative integer k. Let c be a certain candidate that does not win under plurality rule when all the voters of V vote. We are asked whether there exists a set of voters groups $S_a \subseteq I$, $|S_a| \leq k$, such that if we remove from V all the voters in S_a then the plurality winner is c.

Definition 4 *(Defense against group-level election control).* Given a set C of candidates, a set V of voters represented by their preference order, a set I of non-overlapping groups of voters, and non-negative integers k and m. Let ω be the candidate that wins under plurality rule when all the voters of V vote. We are asked whether there exists a set of voters groups $S_d \subseteq I$, $|S_d| \leq m$, such that for any selection of voters groups $S_a \subseteq I$, $S_a \cap S_d = \emptyset$ and $|S_a| \leq k$, if we remove from V all the voters in S_a then the plurality winner is still ω.

[2] This guarantees the existence of an equilibrium, whereas other tie-breaking rules do not. Moreover, this is easy for the defender to ensure, since they can make a small change to the strategy to make the attacker strictly prefer the defender-preferred target while sacrificing minimal utility.

Table 1
Table of complexity.

Case	Complexity
Group-level destructive control	*poly* time (Theorem 1)
Group-level constructive control	NP-complete [18]
Preventing destructive control	NP-complete (Theorem 2)
Preventing constructive control	co-NP-hard (Theorem 3)

We model protection decisions as a Stackelberg game in which the defender (of the elections), acting as a leader, allocates m protection resources among voter groups (e.g., patrols at polling stations or voting districts to be audited), and the attacker, who is a follower, subsequently decides which k groups to attack. Formally, let $s = \langle s_i : i \in \{1, \cdots, n\}\rangle$ denote a pure strategy of the defender, where $s_i \in \{0, 1\}$ indicates whether a voter group i is protected. Similarly, we encode the attacker's pure strategy by a vector a where binary a_i indicates whether group i is attacked. A group i is deleted if and only if it is attacked ($a_i = 1$) and it is not protected ($s_i = 0$). We use $\mathcal{S}$ and $\mathcal{A}$ to represent the (pure) strategy space of the defender and the attacker respectively. Given a pair of strategies (s, a), let $W^c(s, a) \in \{0, 1\}$ be an indicator denoting whether a candidate c wins the election. Notice that the value of $W^c(s, a)$ is easy to compute for a given (s, a): we need only to count the votes in the remaining groups. The defender's utility is then $W^\omega(s, a)$, whereas the utility of the attacker is $-W^\omega(s, a)$ in *destructive control*, and $W^\upsilon(s, a)$ in *constructive control* where the attacker wishes that candidate υ wins.

We allow the defender to commit to a randomized strategy, while the attacker observes the randomization, rather than actual realizations of which voter groups are protected. This is motivated by the fact that attacks typically require significant preparation, and would be based on many empirical observations of protection decisions through reconnaissance activities, such as observing allocation of defensive resources (e.g., audits) from past elections, or just replicating the algorithmic tools used by defenders to compute randomized defense strategies (as much information about these is likely publicly available due to the transparency standards for democratic institutions) [58]. Since the attacker chooses a best response to the defender's strategy, it suffices to consider only pure strategy attacks. Let $\mathbf{x}$ denote the defender's randomized (mixed) strategy, with x_s the probability that a pure strategy $s \in \mathcal{S}$ is used. Given a mixed strategy $\mathbf{x}$ of the defender and a pure strategy a of the attacker, the probability that a particular candidate c wins is $P^c(\mathbf{x}, a) = \sum_{s \in \mathcal{S}} x_s W^c(s, a)$. Thus, the expected utility of the defender is always $P^\omega(\mathbf{x}, a)$, while the attacker's expected utility is $-P^\omega(\mathbf{x}, a)$ for destructive control and $P^\upsilon(\mathbf{x}, a)$ for constructive control. In the case of destructive control, the players are engaged in a zero-sum game, where the Stackelberg equilibrium strategy for the defender is equivalent to its Nash equilibrium [37]. In constructive control, however, the game is non-zero-sum. The distinction between destructive and constructive control, therefore, has significant algorithmic ramifications in our setting, as will be seen below.

5. Complexity of control and protection

Our first task is to consider the computational complexity of group-level election control, which has been only partially addressed in prior literature, and the complexity of protecting elections from control. Our results are summarized in Table 1.

5.1. Complexity of group-level destructive control

It is well known that constructive and destructive control of plurality by deleting individual voters can be computed in polynomial time [4,27]. Allowing the attacker to select specific groups may appear to significantly complicate the problem. Indeed, Erdélyi et al. [18] showed that group-level constructive control (i.e., making candidate υ win by deleting voter groups) is NP-complete even with plurality. Surprisingly, we now show that the destructive variant of group-level control can still be computed in polynomial time, significantly generalizing the previous result of Hemaspaandra et al. [27]. Intuitively, control succeeds as long as there exists a candidate $c \in C$ who has at least as many votes as ω after k groups are removed. Consequently, the attacker can consider one candidate c at a time, checking if k groups can be deleted so that c has a higher vote count than ω. Moreover, if we fix $c \in C$, it is easy to check whether it is possible to get more votes for c than ω: we would just delete the k groups in which ω is most favored over c.

Formally, let $\mathbf{t}^{c-\omega} = \langle t_i^{c-\omega} : i \in I\rangle$ be a vector with $t_i^{c-\omega} = t_{ic} - t_{i\omega}$, that is, the vote advantage of c over ω in group $i \in I$. For a vector $\mathbf{t}^{c-\omega}$, define $sum(\mathbf{t}^{c-\omega}) = \sum_i t_i^{c-\omega}$. Then, $sum(\mathbf{t}^{c-\omega})$ is the total difference of votes between c and ω. For example, suppose that $\mathbf{t}^{c-\omega}$ is $\langle -3, -2, 1\rangle$. This means that ω has more votes than c in the first two groups, but fewer (by 1) in the third. If the attacker can attack 2 groups, he will succeed by attacking the first two, leading c to have 1 more vote left than ω. The following proposition shows that it is, in fact, sufficient to delete k groups with smallest $t_i^{c-\omega}$ to verify whether it is possible to make c have a larger vote count than ω. For convenience, define $\mathbf{t}^{c-\omega} \setminus k$ to be the portion of the vector $\mathbf{t}^{c-\omega}$ remaining after the k groups with smallest $t_i^{c-\omega}$ have been deleted.

Proposition 1. *For a given candidate $c \in C$, it is possible to delete k groups to ensure that $t_c > t_\omega$ if and only if $sum(\mathbf{t}^{c-\omega} \setminus k) > 0$.*

Proof. The $\Leftarrow$ direction is straightforward: if $sum(\mathbf{t}^{c-\omega} \setminus k) > 0$, then by definition of $\mathbf{t}^{c-\omega} \setminus k$ we have accomplished our goal and $t_c > t_\omega$. For the $\Rightarrow$ direction, if deleting the smallest k elements in $\mathbf{t}^{c-\omega}$ still leaves $sum(\mathbf{t}^{c-\omega} \setminus k) < 0$, then it is impossible to find any other subset of k groups $G \subseteq I$ to delete and have $t_c > t_\omega$, since we chose the k groups with the largest $t_{i\omega} - t_{ic}$, and, consequently, added the largest possible $\sum_i t_{i\omega} - t_{ic}$ to $sum(\mathbf{t}^{c-\omega})$. Since the remaining tally difference is still negative, it is not possible to make c have more votes than ω. □

Algorithm 1: Optimal Election Control by Group Deletion.

```
1 for $c \in C^{-\omega}$ do
2     $\mathbf{t}^{c-\omega} \setminus k \leftarrow$ Sort $\mathbf{t}^{c-\omega}$ in ascending order, delete the first $k$ elements in $\mathbf{t}^{c-\omega}$;
3     if $sum(\mathbf{t}^{c-\omega} \setminus k) > 0$ then
4         return Attack voter groups corresponding to deleted elements;
5 return No control approach;
```

The complete approach for computing a group-level election control is given in Algorithm 1. For each candidate $c \in C \setminus \{\omega\}$, denoted by $C^{-\omega}$, Lines 1–4 check whether there exists an attack where c beats ω (based on Proposition 1). If no such attack exists for all candidates in $C^{-\omega}$, election control is not possible. The complexity of Algorithm 1 is $O(|C|n \log n)$, which yields the following result:

Theorem 1. *Destructive election control by deleting k voter groups can be accomplished in $O(|C|n \log n)$ time.*

5.2. Complexity of preventing destructive control

To show the complexity of preventing destructive control, we use a reduction from a known NP-complete problem, the hitting set problem, which is defined as follows.

Definition 5 *(Hitting set problem).* **Given:** A set G and a set U consisting of subsets $\hat{G}$ of G. **Question:** Does there exist a 'hitting set' $G' \subseteq G$ with $|G'| = m$, so that $\forall \hat{G} \in U, G' \cap \hat{G} \neq \emptyset$.

Theorem 2. *Checking whether m protection resources are sufficient to prevent destructive control is NP-complete.*

Proof. Based on Theorem 1, our problem is in NP. To show that it is NP-hard, we reduce from the hitting set problem. Specifically, we show that for any instance of the hitting set problem, we can construct an election with n voter groups, so that there exists a hitting set G' if and only if it is possible to prevent any control with m resources, i.e., the attacker cannot make ω lose by attacking any subset of the $n - m$ unprotected groups.

Given an instance of the hitting set problem, we construct an election as follows. There are $n = |G|$ voter groups and $|U| + 1$ candidates. Each $i \in G$ corresponds to a voter group. Each $\hat{G} \in U$ can be considered as a label of a specific candidate other than ω. For candidate c with label $\hat{G}$, for any voter group i, if $i \in \hat{G}$, then let $t_i^{c-\omega} = -1$, i.e., c has 1 fewer vote than ω in group i; otherwise let $t_i^{c-\omega} = 0$, i.e., c and ω are tied in group i. Assume that up to $k = n - m$ groups are attacked. For example, let $G = \{1, 2, 3\}$ and $U = \{\{1, 2\}, \{2, 3\}, \{2\}\}$. Then the vote states of the candidates are shown in Table 2.

The $\Leftarrow$ direction: If there exists a defender strategy which protects m voter groups, i.e., $G' \subset G$ with $|G'| = m$, so that the attacker has no way to control the election, it indicates that for each candidate c, i.e., an element $\hat{G} \in U$, at least one voter group i in which $t_i^{c-\omega} = -1$ is protected, i.e., $G' \cap \hat{G} \neq \emptyset$. This is because if there exists a candidate c, all voter groups with $t_i^{c-\omega} = -1$ are unprotected, then the attacker can successfully attack all such groups and c will be tied with ω among the remaining votes. Thus the protected voter groups satisfy $\forall \hat{G} \in U, G' \cap \hat{G} \neq \emptyset$, which is the required hitting set.

The $\Rightarrow$ direction: Given a hitting set $G' \subset G$, the defender can protect all voter groups $i \in G'$. Thus, even if the attacker attacks all the unprotected voter groups, each candidate $c \in C^{-\omega}$ still has at least 1 vote fewer than ω. Therefore, no attacker strategy can control the election. □

5.3. Complexity of preventing constructive control

To show the complexity of preventing constructive control, we use a reduction from a known NP-complete problem, the set cover problem, which is defined as follows.

Table 2
Example votes.

Candidates Label	1 $\{1, 2\}$	2 $\{2, 3\}$	3 $\{2\}$	ω
Group 1	0	1	1	1
Group 2	0	0	0	1
Group 3	1	0	1	1

Table 3
Example votes in constructed election.

Candidates	1	2	3	υ	Label
Group 1	0	0	1	0	$\{1, 2\}$
Group 2	1	0	1	0	$\{2\}$
Group 3	1	1	0	0	$\{3\}$
Groups 4, 5	1	1	1	0	
Groups 6–9	0	0	0	1	
Group 10	1	0	1	0	

Definition 6 *(Set cover problem).* **Given:** A set G and a set U consisting of subsets $\hat{G}$ of G. **Question:** Does there exist a 'set cover' $U' \subseteq U$ with $|U'| = r$, so that the union of U' is G.

Theorem 3. *Checking whether m protection resources are sufficient to prevent constructive control is co-NP-hard.*

Proof. We prove Theorem 3 by showing that for any instance of the set cover problem, we can construct an election such that there is a set cover if and only if m protection resources is not sufficient to prevent constructive control.

Assume that $|G| = l$ and $|U| = n$. We construct an election in which there are $l + 1$ candidates and $n + r + 2m$ voter groups. m is the number of protection resources and r is the size of the 'set cover' $U' \subseteq U$. Each $c \in G$ corresponds to a candidate except the $(l + 1)$st candidate. Elements in U correspond to the first n voter groups, i.e., each $\hat{G} \in U$ can be considered as the label of a voter group.

We now consider the votes from each group. For the first n groups, if a group is labeled by $\hat{G}$, then there is one vote from this group to each candidate $c \in G \setminus \hat{G}$ and no vote to other candidates. For the $(n+1)$st group to $(n+m)$th group, there is one vote for each candidate in G in each group. For the remaining $r + m$ groups, each group contains only one voter who votes for the $(l + 1)$st candidate. The attacker's goal is to have the $(l + 1)$st candidate win, i.e., the $(l + 1)$st candidate is υ. If υ is not the original winner in the voting set above, we assume that the attacker can delete $k = n - r$ groups. Otherwise, we first choose candidates $c^* \in G$ who have the most votes among candidates in G, then add an 'extra group' which has $t_\upsilon - t_{c^*} + 1$ votes for candidates c^*, and no vote for any other candidate.[3] In this case, we assume that the attacker can delete $k = n - r + 1$ groups. For example, assume that $G = \{1, 2, 3\}$, $U = \{\{1, 2\}, \{2\}, \{3\}\}$, $r = 2$ and $m = 2$. Then there are 4 candidates and 10 groups. The 10th group is added since no candidate beats υ in the first 9 groups. Votes for candidates by voter group are shown in Table 3.

Next, we prove the theorem from both directions.

The ⇒ direction: If there exists an r set cover U', then m resources are not sufficient to protect the election.

Indeed, if there exists an r set cover U', then one attacker strategy which can make υ win is to attack the $n - r$ groups labeled with $i \in U \setminus U'$ (plus the extra group if it is added; in the example above, the attacker can attack groups 2 and 10). In the remaining $2r + 2m$ groups, υ will have more votes than any other candidate. This indicates that if the defender wants to completely block the attacker, at least one group labeled with $i \in U \setminus U'$ or the extra group, e.g., group i, should be protected. Thus at least one of the m groups in which there is one vote for each candidate in G, e.g., group j, cannot be protected. In this case, the attacker can keep his strategy, except for attacking target j instead of i, and still let υ win. In the above example, to completely block the attacker, the defender needs to protect group 2. This indicates that one of groups 4 and 5 cannot be protected. Thus the attacker can turn to attack the unprotected group and let υ win. Therefore, if there exists an r set cover then m resources are not sufficient to protect the election.

The ⇐ direction: If there does not exist an r set cover, then m resources are sufficient to protect the election.

This is because the defender can protect the m groups in which there is one vote for each candidate in G (groups 4 and 5 in the example). The attacker will not attack the groups in which only υ has votes, and he will attack the $n - r$ groups labeled with $i \in U$ (plus the extra group if it is added). Since there does not exist an r set cover, in the remaining r groups labeled with $i \in U$, at least one candidate will have r votes, and thus this candidate will at least tie with υ, and the attacker cannot succeed. Therefore, if there does not exist an r set cover, then m resources are sufficient to protect the election. Equivalently, if m resources are not sufficient to protect the election, then there exists an r set cover. □

6. Optimal defense against election control

The results thus far appear rather negative from the defender's perspective: group-level control is computationally hard to prevent. However, these were worst-case results. Next, we turn to the more pragmatic question of developing practical algorithmic approaches for protection against election control. We do so by modeling the problem as a Stackelberg security game between the defender and the attacker, as is introduced in Section 3. In our scenario, given a pair of the defender's mixed strategy and the attacker's pure strategy, i.e., $(\mathbf{x}, a)$, the defender's utility is $u^D(\mathbf{x}, a) = P^\omega(\mathbf{x}, a)$. The attacker's utility, on the other hand, depends on the form of control they wish to exercise. In destructive control settings,

[3] We can add one more vote to one of c^* in the extra group to avoid tie in the game. This does not affect the following proof.

$u^A(\mathbf{x}, a) = -P^\omega(\mathbf{x}, a)$, whereas for constructive control, $u^A(\mathbf{x}, a) = P^\upsilon(\mathbf{x}, a)$. We adopt the Strong Stackelberg Equilibrium (SSE) as the solution concept, which is formally defined by Definition 1. Next, we consider defending each type of election control respectively.

6.1. Defense against destructive control

If the attacker's goal is to perform destructive control, the defender and the attacker have exactly opposing objectives, and the game is therefore zero-sum. Since SSE and Nash equilibria are equivalent in zero-sum games [37], we can use a well-known linear programming formulation for solving zero-sum normal-form games [12]. We call this formulation Core-LP.

$$\text{Core-LP}(\mathcal{S}, \mathcal{A}): \quad \max_{\mathbf{x}, p} \quad p \tag{1a}$$

$$p \le \sum\nolimits_{s \in \mathcal{S}} x_s P^\omega(s, a), \quad \forall a \in \mathcal{A}. \tag{1b}$$

As introduced in previous sections, $\mathbf{x} = \{x_s : s \in \mathcal{S}\}$ denotes a mixed strategy of the defender, with x_s representing the probability that strategy s is played. Core-LP results in a mixed strategy of the defender which maximizes the defender utility given that the attacker responds the best. The central challenge with this approach is that it requires one to explicitly enumerate all pure strategies for both the defender and attacker. Since in our case the strategy space for both players is exponential, this is a non-starter. We therefore develop a *Double Oracle* approach for addressing this scalability issue.

The double oracle framework is a common approach for solving zero-sum games in which both players have exponential strategy spaces [47,34]. The idea is to start with a small set of strategies for both players, compute equilibrium in this restricted game using Core-LP, and check whether either player has a best response in the full strategy space that improves their payoff. If such a strategy exists for either player, it is added to the Core-LP, which is solved again. Otherwise, we have proven that the resulting restricted equilibrium is a Stackelberg/Nash equilibrium of the full game.

The Double-Oracle approach is not itself an algorithm, as it does not specify how to compute a best response for each player in the full strategy space. Indeed, in general this would require full enumeration of player strategies. The key is to develop effective approaches to compute such best responses—that is, effective oracles for both players—which is problem specific. For example, none of the prior approaches (e.g., [34]) are applicable in our case, because of modeling differences. Our central contributions in this section are therefore: 1) novel mixed-integer linear programming (MILP) formulations for both oracles, and 2) heuristic algorithms to speed up the computation of the oracles.

Our full double-oracle method is shown in Algorithm 2. Line 3 computes the mixed strategy equilibrium of the restricted game, $(\mathbf{x}, \mathbf{y})$, where $\mathbf{y}$ is the dual solution of Core-LP representing attacker's mixed strategy. We then make use of two types of oracles: heuristic oracles, which allow us to quickly check the existence of *better* responses (AO-Better and DO-Better, for attacker and defender, respectively), and exact oracles (AO-MILP and DO-MILP), which are optimal.

Algorithm 2: Double Oracle Approach.

```
1  Input: $\mathcal{S}' \subset \mathcal{S}$; $\mathcal{A}' \subset \mathcal{A}$;
2  while do
3      $(\mathbf{x}, \mathbf{y}) \leftarrow$ Core-LP$(\mathcal{S}', \mathcal{A}')$;
4      $a \leftarrow$ AO-Better$(\mathbf{x})$;
5      if $a = \emptyset$ then $a \leftarrow$ AO-MILP$(\mathbf{x})$;
6      $s \leftarrow$ DO-Better$(\mathbf{y})$;
7      if $s = \emptyset$ then $s \leftarrow$ DO-MILP$(\mathbf{y})$;
8      if $a \in \mathcal{A}'$ and $s \in \mathcal{S}'$ then
9          return $\mathbf{x}$;
10     else
11         $\mathcal{A}' \leftarrow \mathcal{A}' \cup \{a\}, \mathcal{S}' \leftarrow \mathcal{S}' \cup \{s\}$;
```

Next, we describe both the exact and heuristic oracles for the attacker and defender, observing in the process that both best response problems are NP-complete.

6.1.1. Attacker oracle

Complexity. In Theorem 1 we had already shown that destructive control in our model can be performed in polynomial time. However, this result assumed that no protection is deployed, or, equivalently, that protection is deterministic. Surprisingly, when protection is randomized, destructive control, which we also refer to as the attacker's *best response* or *oracle*, is NP-complete, as the following result attests (in this result, $\mathcal{S}'$ represents the support of the defender's mixed strategy).

Theorem 4. *Let $\mathcal{S}'$ be a set of defender strategies. Checking whether there exist k groups an attack on which would control an election no matter which $s \in \mathcal{S}'$ is played by the defender is NP-complete, even with only two candidates.*

Proof. Clearly, given k groups that will be attacked, it is easy to verify that the defender will not be able to prevent the control by playing any $s \in \mathcal{S}'$, and thus the problem is in NP. We prove hardness by a reduction from the hitting set problem given in Definition 5. Specifically, we show that for any instance of the hitting set problem, we can construct an election with n voter groups, 2 candidates, and a set $\mathcal{S}'$ of defender strategies, so that there exists a hitting set G' if and only if there exists an attacker strategy which can control the election no matter which defender strategy $s \in \mathcal{S}'$ is played.

Given an instance of a hitting set problem, we construct an election with $|G|+1$ voter groups and two candidates, ω and some other candidate c, and assume that the attacker can attack m groups. Each $i \in G$ corresponds to a voter group, in which we assume that $t_i^{c-\omega} = -1$, i.e., c has one fewer vote than ω in voter group i. In the extra voter group j which does not correspond to any element in G, we assume that $t_j^{c-\omega} = |G| - 1$. Thus c has 1 less vote than ω in total. Each $\hat{G} \in U$ can be considered as a label of a defender's pure strategy, in which voter group j and voter groups $i \in G \setminus \hat{G}$ are protected.

For example, assume that $G = \{1, 2, 3\}$ and $U = \{\{1, 2\}, \{3\}\}$. This indicates that there are 4 voter groups, i.e., 3 correspond to items in G and one extra voter group; the defender has two pure strategies in the support set, each is labeled by one item in U. In the first three voter groups c has 1 fewer vote than ω, while in the last group c has 2 more votes than ω. In the defender strategy labeled by $\{1, 2\} \in U$, groups 3 and 4 are protected. In the defender strategy labeled by $\{3\} \in U$, groups 1, 2 and 4 are protected.

The $\Leftarrow$ direction: If there exists an attacker strategy which attacks m voter groups i.e., $G' \subset G$ with $|G'| = m$, so that he can control the election no matter which defender strategy $s \in \mathcal{S}'$ is played, it indicates that given the defender strategy s labeled by $\hat{G} \in U$, at least one voter group i with $i \in \hat{G}$ and $t_i^{c-\omega} = -1$ is attacked. Thus for each $\hat{G} \in U, G' \cap \hat{G} \neq \emptyset$. Otherwise the attacker cannot control the election if s is played. Therefore, G' is a required hitting set.

Still consider the previous example. If $|m| = 2$, the attacker can attack voter groups $G' = \{2, 3\}$, thus he can control the election no matter which pure strategy the defender plays. Correspondingly, G' is a required hitting set.

The $\Rightarrow$ direction: Given a hitting set $G' \subset G$ with $|G'| = m$, the attacker can attack all voter groups $i \in G'$. Thus no matter which $s \in \mathcal{S}'$ is played by the defender, at least one unprotected voter group with $t_i^{c-\omega} = -1$ is attacked. Since ω only has 1 more vote than c in the original voting, the attacker can prevent ω from winning no matter which $s \in \mathcal{S}'$ is played. □

Hardness of destructive control when the defender randomizes has an interesting theoretical implication in the context of the traditional approach where resistance of elections to control is identified with computational hardness of the latter [14]. The fact that plurality is easy to control for a deterministic (or no) defense, but hard when protection is randomized would itself be considered a significant reduction in the vulnerability of the election to control. Below, we go a step further.

Exact solution. Although computing the attacker's best response (oracle) is NP-complete, we now develop an exact compact mixed-integer linear program (MILP) for it, which we term **AO-MILP**. Formally, the attacker's best response involves minimizing the probability that ω wins, i.e., solving $\min_{a\in\mathcal{A}} \sum_{s\in\mathcal{S}'} x_s W^\omega(\mathbf{x}, a)$ for a given mixed strategy $\mathbf{x}$ with the support set $\mathcal{S}'$. Our first step is to formulate the attacker oracle as a mathematical (non-linear) program. The main technical challenge involved is representing $W^\omega(s, a)$, which is a non-trivial function of s and a. We do this implicitly in AO-MILP by using an auxiliary binary variable z_s.

$$\text{AO-MILP}: \quad \min_{a_i, z_s, e_s^c \in \{0,1\}} \quad \sum_{s\in\mathcal{S}'} \quad (1 - z_s) \cdot x_s \tag{2a}$$

$$\sum_i a_i \le k \tag{2b}$$

$$\sum_{c\in C^{-\omega}} e_s^c = 1, \quad \forall s \in \mathcal{S}' \tag{2c}$$

$$z_s \sum_{c\in C^{-\omega}} e_s^c \left(\sum_i t_i^{c-\omega}(1 - (1 - s_i)a_i)\right) \ge 0, \forall s \in \mathcal{S}'. \tag{2d}$$

Constraint (2b) enforces feasibility of the attacker's strategy vector a. Next we explain Constraints (2c)–(2d). Given a strategy pair (s, a), votes in group i are deleted only if $s_i = 0$ and $a_i = 1$. Thus for each candidate $c \in C^{-\omega}$, the vote difference between c and ω is $sum(\mathbf{t}^{c-\omega} \setminus k) = \sum_i t_i^{c-\omega} - \sum_i (1 - s_i) a_i t_i^{c-\omega}$. Note that $z_s = 1$, i.e., ω loses given strategy pair (s, a), as long as there exists one candidate who has no fewer votes left than ω given (s, a), i.e., $sum(\mathbf{t}^{c-\omega} \setminus k) \ge 0$. Variables e_s^c are thus introduced to check whether there exists such a candidate. Constraints (2c), (2d), and the objective together ensure that if there exists such a candidate c^* for some s, the corresponding $e_s^{c^*}$ will be set as 1 and e_s^c for all other candidates will be set as 0. Thus, $\sum_{c\in C^{-\omega}} e_s^c \left(\sum_i t_i^{c-\omega}(1 - (1 - s_i)a_i)\right) \ge 0$, and the associated $z_s = 1$, yielding, in combination with Constraint (2b) a pure strategy for the attacker that minimizes the probability that ω wins given $\mathbf{x}$. While AO-MP includes non-linear constraint (2d), because all variables involved are binary, this constraint can be linearized in a standard way using McCormick inequalities [46], yielding an MILP for computing the attacker's best response. Specifically, we introduce two new variables $w_s^c = z_s e_s^c$ and $v_{s,i}^c = w_s^c a_i$, so that constraint (2d) becomes linear related to w^c and $v_{s,i}^c$. We then add the following constraints, so that the resulted linear program is equivalent to the original program.

$$w_s^c \le z_s, w_s^c \le e_s^c, \quad \forall c, s \tag{3a}$$

$$w_s^c \ge z_s + e_s^c - 1, \quad \forall c, s \tag{3b}$$

$$v^c_{s,i} \le w^c_s,\, v^c_{s,i} \le a_i, \quad \forall c, s, i \in \{1, \cdots, n\} \tag{3c}$$

$$v^c_{s,i} \ge w^c_s + a_i - 1, \quad \forall c, s, i \in \{1, \cdots, n\} \tag{3d}$$

$$w^c_s,\, v^c_{s,i} \in \{0, 1\} \tag{3e}$$

Heuristic "better" response. The main issue with AO-MILP is poor scalability. However, we need only compute a *better* response for the attacker in each iteration of the Double-Oracle method to make progress; by doing so quickly, even if heuristically, we can considerably speed up equilibrium computation. As long as we ultimately fall back on the MILP to check optimality, we can still guarantee that the final double-oracle solution is an SSE.

We take two steps to find a better response for the attacker. First, we look for a subset $\mathcal{S}'' \subset \mathcal{S}'$ with $\sum_{s\in\mathcal{S}''} x_s > 1 - p$, where p is the objective value of Core-LP restricted to a small subset of attacker strategies $\mathcal{A}'$ in the previous iteration, i.e., the probability that ω still wins the game given the current best attacker strategy. Second, we look for an attacker pure strategy a which can make ω lose no matter which pure strategy $s \in \mathcal{S}''$ is played by the defender, i.e., $W^\omega(s, a) = 0$ for each $s \in \mathcal{S}''$. If we can successfully find such a set $\mathcal{S}''$ and a pure strategy a, the attacker will make ω lose with probability at least $\sum_{s\in\mathcal{S}''} x_s$ (win with a probability less than p) if he plays pure strategy a. Thus, a is a better strategy than any $a' \in \mathcal{A}'$.

The full heuristic approach, **AO-Better**, is shown in Algorithm 3. We first sort the defender strategies in $\mathcal{S}'$ in decreasing order of x_s, obtaining a sorted vector $\bar{S}$ with s^ρ the ρth largest element (Line 3). We then look for set $\mathcal{S}''$ consisting of adjacent strategies in $\bar{S}$ (Lines 5–6). For each $\mathcal{S}''$, we check if there exists a candidate c, such that if the attacker attacks k areas which are not protected by any strategy $s \in \mathcal{S}''$, c will have more votes remaining than ω. If there exists such a candidate, then the corresponding attacker strategy leads ω to lose no matter which $s \in \mathcal{S}''$ is played by the defender, and is better than any in $\mathcal{A}'$ (Lines 8–11). If no better strategy is found, then **AO-Better** returns an empty set.

Algorithm 3: Attacker's Better Response (**AO-Better**).

```
1  input: $\mathcal{S}'$, $\mathbf{x}$, $p$;
2  $\bar{S} = \langle s^\rho, \rho \in 1, 2, 3, \cdots \rangle \leftarrow$ sort $s \in \mathcal{S}'$ by decreasing $x_s$;
3  for $\rho$ in $1..|\bar{S}|$ do
4      $p' \leftarrow x_{s^\rho}$, $\mathcal{S}'' \leftarrow \{s^\rho\}$, $\rho' \leftarrow \rho + 1$;
5      while $p' \le 1 - p$ and $\rho' \le |\bar{S}|$ do
6          $p' \leftarrow p' + x_{s_{\rho'}}$, $\mathcal{S}'' \leftarrow \mathcal{S}'' \cup \{s^{\rho'}\}$, $\rho' \leftarrow \rho' + 1$;
7      if $p' > p$ then
8          for $c \in C^{-\omega}$ do
9              $\mathbf{t}^{c-\omega} \setminus k \leftarrow$ delete $k$ elements in $\mathbf{t}^{c-\omega}$ with the smallest $t_i^{c-\omega}$ and $s_i = 0$;
10             if $sum(\mathbf{t}^{c-\omega} \setminus k) \ge 0$ then
11                 return attack the $k$ groups corresponding to deleted elements;
12 return $\emptyset$;
```

6.1.2. Defender oracle

Hardness of the defender oracle follows from Theorem 2. We now proceed to develop a mathematical programming approach for solving it in practice.

Exact solution. The defender's oracle, or best response, is to maximize the probability that ω wins given a mixed strategy of the attacker $\mathbf{y} = \langle y_a : a \in \mathcal{A}' \rangle$, i.e., to solve $\max_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}'} W^\omega(s, a) y_a$. Just as in the attacker oracle formulation, we proceed to develop the (non-linear) mathematical integer program to compute the defender's best response.

$$\text{DO-MILP:} \qquad \max_{s_i, z_a \in \{0,1\}} \sum_{a\in\mathcal{A}'} z_a \cdot y_a \tag{4a}$$

$$\sum_i s_i \le m \tag{4b}$$

$$z_a \sum_i (t_i^{c-\omega}(1-(1-s_i)a_i)+1) \le 0, \forall c \in C^{-\omega}, a \in \mathcal{A}'. \tag{4c}$$

There is an important difference in this formulation from the attacker oracle: in particular, $z_a = 1$ (that is, the defender successfully blocks an attack strategy $a \in \mathcal{A}'$, so that $W^\omega(s, a) = 1$, where $\mathcal{A}'$ is the attacker strategy from the previous iteration of Double-Oracle) only if all candidates $c \in C^{-\omega}$ have fewer votes remaining than ω. Constraint (4c) ensures that $z_a = 1$ only when $\forall c \in C^{-\omega}$, $\sum_i t_i^{c-\omega} - \sum_i (1 - s_i) a_i t_i^{c-\omega} < 0$, while Constraint (4b) enforces feasibility of the defender's strategy. The resulting DO-MILP thereby chooses the defender strategy which maximizes the probability that ω wins given a fixed attacker mixed strategy $\mathbf{y}$. We can then linearize the nonlinear constraint (4c) as before by using McCormick inequalities [46], obtaining an MILP formulation of the defender oracle.

Heuristic "better" response. We first look for a subset $\mathcal{A}'' \subset \mathcal{A}'$ with $\sum_{a \in \mathcal{A}''} y_a > p$ (similar to how we look for $\mathcal{S}''$ in the attacker oracle). Then we look for a defender pure strategy s which can "block" all attacker strategies $a \in \mathcal{A}''$, ensuring that ω will win with probability higher than p. If such a strategy is found, then it is a better response for the defender. Algorithm 4 presents the full heuristic procedure.

Algorithm 4: Defender Oracle with Better Response.

```
$s = \langle s_i = 0 : \forall i \in \{1, \cdots, n\} \rangle$;
for $\mathcal{A}''$ with $\sum_{a \in \mathcal{A}''} y_a > p$ do
    res $\leftarrow$ 0 ;
    for $c \in C^{-\omega}$ do
        $\mathbf{t}' \leftarrow \langle t_i^{c-\omega} : i$ with $a_i = 0, \forall a \in \mathcal{A}''$ or $s_i = 1 \rangle$;
        while $sum(\mathbf{t}') \geq 0$ and $res < m$ do
            $\mathbf{t}'' \leftarrow \mathbf{t}^{c-\omega} \setminus \mathbf{t}', i^* \leftarrow \mathrm{argmin}_i \{\mathbf{t}''\}$;
            $\mathbf{t}' \leftarrow \mathbf{t}' \cup \{t_{i^*}^{c-\omega}\}$, $s_{i^*} \leftarrow 1$, res $\leftarrow$ res + 1;
    if $\forall c \in C^{-\omega}, sum(\mathbf{t}') < 0$ then
        return $s$;
return $\emptyset$;
```

6.2. Defense against constructive control

Having addressed defense against destructive control, we now turn to the issue of constructive control.

6.2.1. Heuristic algorithm

Before addressing equilibrium strategies in the Stackelberg security game, we first introduce a heuristic algorithm which computes a pure strategy to prevent constructive control in certain cases. While such strategies are generally hard to compute (as is investigated by Theorem 3), we now show that it can be done in polynomial time in an important restricted setting. Assume that the attacker's goal is to let candidate υ win the game. One defender strategy which can prevent the attacker is to protect m voter groups, so that a certain candidate $c \in C$ has more votes than υ despite how the unprotected groups are deleted. We call such a strategy *c-dominant-strategy*. A c-dominant-strategy makes the attacker fail and she would be possibly deterred from attacking. Since it is computationally hard for the defender to ensure that ω wins, she can at least find a c-dominant-strategy efficiently, if one exists. Later experiments will show that such strategies exist in a considerable proportion of the games, especially the ones with higher (defense resource/voter group) ratio.

Now, if we fix $c \in \mathcal{C}$, it is easy to check whether there exists a c-dominant-strategy: we can protect the m groups in which c is most favored over υ, and check whether the attacker can have υ beat c by deleting k of the unprotected groups which favor c more than υ. Similar as previous definition of $\mathbf{t}^{c-\omega}$, let $\mathbf{t}^{c-\upsilon} = \langle t_i^{c-\upsilon} : i \in I \rangle$ be a vector with $t_i^{c-\upsilon} = t_{ic} - t_{i\upsilon}$, that is, the vote advantage of c over υ in group $i \in I$. For a vector $\mathbf{t}^{c-\upsilon}$, define $sum(\mathbf{t}^{c-\upsilon}) = \sum_i t_i^{c-\upsilon}$. Then, $sum(\mathbf{t}^{c-\upsilon})$ is the total difference of votes between c and υ. The following proposition gives the condition for determining the existence of a c-dominant-strategy. For convenience, let e^c be the set of groups corresponding to the $(m+1)$st largest to the $(m+k)$th largest $t_i^{c-\upsilon}$ with $t_i^{c-\upsilon} > 0$. e^c is an empty set if the group with the $(m+1)$st largest $t_i^{c-\upsilon}$ has $t_i^{c-\upsilon} < 0$. Let $\mathbf{t}^{c-\upsilon} \setminus e^c$ be the portion of the vector $\mathbf{t}^{c-\upsilon}$ remaining after elements corresponding to groups in e^c are deleted. For example, assume that there exists a candidate c with $\mathbf{t}^{c-\upsilon} = \langle 10, 5, -5, -3 \rangle$, and that $m = 1$ and $k = 2$, then e^c includes only the group corresponding to 5 (since the group with the 3rd largest $t_i^{c-\upsilon}$ has $t_i^{c-\upsilon} = -3 < 0$) and $\mathbf{t}^{c-\upsilon} \setminus e^c = \langle 10, -5, -3 \rangle$.

Proposition 2. *For candidate $c \in \mathcal{C}$, there exists a c-dominant-strategy if and only if $sum(\mathbf{t}^{c-\upsilon} \setminus e^c) > 0$.*

Proof. First, given an allocation of security resources which protects a set ψ of m groups, to let υ beat c, the attacker's best response is to attack unprotected groups to suppress the advantage of c over υ the most, i.e., a group $e^* = \mathrm{argmax}_{e, e \subset I \setminus \psi, |e| \leq k}\, sum(e)$ (without loss of generality, here $sum(e) = sum(\{t_i^{c-\upsilon} : i \in e\})$).

For the $\Leftarrow$ direction, if $sum(\mathbf{t}^{c-\upsilon} \setminus e^c) > 0$, a c-dominant strategy is to protect m groups with the m largest $t_i^{c-\upsilon}$. Thus $e^* = e^c$ and no attacker strategy can let υ beat c.

For the $\Rightarrow$ direction, if there exists a c-dominant-strategy which protects groups in ψ, then $sum(e^*) \geq sum(e^c)$. Thus protecting the m groups with the m largest $t_i^{c-\upsilon}$ is also a c-dominant-strategy and $sum(\mathbf{t}^{c-\upsilon} \setminus e^c) > 0$. □

Based on Proposition 2, we propose Algorithm 5, which returns a c-dominant strategy in polynomial time if one exists.

6.2.2. SSE

The key new challenge in the constructive control setting is that the resulting Stackelberg game is no longer zero-sum, and we therefore cannot use the double-oracle approach directly. The SSE strategies can be computed through a mixed integer linear program. We first introduce a variable $q_a \in \{0, 1\} (\forall a \in \mathcal{A})$ to indicate whether the attacker plays pure strategy

Algorithm 5: Greedy Heuristic Algorithm.

```
for $c \in C$ do
    $\mathbf{t}^{c-\upsilon} \setminus e^c \leftarrow$ Delete the set of groups with the $(m+1)$st largest to the $(m+k)$th largest $t_i^{c-\upsilon}$ and $t_i^{c-\upsilon} > 0$;
    if $sum(\mathbf{t}^{c-\upsilon} \setminus e^c) > 0$ then
        return Protect voter groups corresponding to the largest $m$ elements in $\mathbf{t}^{c-\upsilon}$
return No $c$-dominant-strategy exits
```

a ($q_a = 1$) or not ($q_a = 0$). Let M be a large number. The program can be formulated as is shown by the following SSE-MILP. Constraint (5b) places an upper bound of $u^D(\mathbf{x}, a)$ on p^D, but only for the played attacker strategies. Since the objective maximizes p^D, it implies that the defender plays a best response. Similarly, Constraint (5c) forces the attacker to select the best response given $\mathbf{x}$. The first part, $p^A - u^A(\mathbf{x}, a) \geq 0$, implies that p^A must be at least as large as the maximal payoff of playing any other attacker strategy. The second part forces $p^A - u^A(\mathbf{x}, a) \leq 0$ for the played strategies, which will be violated if the attacker does not play a best response. Taken together, the program results in SSE strategies for both players.

$$\text{SSE-MILP}: \quad \max_{\mathbf{x}, p} \quad p^D \tag{5a}$$

$$p^D - u^D(\mathbf{x}, a) \leq (1 - q_a)M, \quad \forall a \in \mathcal{A} \tag{5b}$$

$$0 \leq p^A - u^A(\mathbf{x}, a) \leq (1 - q_a)M, \quad \forall a \in \mathcal{A} \tag{5c}$$

6.2.3. Zero-sum approximation

The MILP above requires us to explicitly enumerate all pure strategies to solve the game, which makes it inefficient for solving large games. While a double-oracle approach cannot be applied here to compute the optimal solution for the defender because the game is not zero-sum, we now show that we can use it to obtain an *approximate* solution to the problem.

First, consider the zero-sum game of defense against destructive control. We call it *ω-zero-sum* game for convenience. Let $(\mathbf{x}^\omega, a^\omega)$ be the equilibrium strategies for the defender and the attacker in this zero-sum game, and let p_0^ω be probability that ω wins the game given $(\mathbf{x}^\omega, a^\omega)$. Note that even when the attacker's goal is to let candidate υ win, if the defender plays $\mathbf{x}^\omega$ and the attacker responds optimally by choosing a strategy $a_\upsilon^\omega = g(\mathbf{x}^\omega)$, where $a_\upsilon^\omega = \text{argmax}_a P^\upsilon(\mathbf{x}^\omega, a)$, the probability that ω wins given $(\mathbf{x}^\omega, a_\upsilon^\omega)$ will be no less than p_0^ω. This is because playing a_υ^ω indicates that the attacker deviates from the optimal strategy in the zero-sum game, which makes the defender better off. Consequently, the solution of this zero-sum game provides a lower bound on the defender's optimal utility against constructive control.

Though the solution to the ω-zero-sum game guarantees a lower bound, we still wish to know how much it differs from the optimal defender strategy. This can be achieved by solving another zero-sum game, which we term the *υ-zero-sum* game. This game is identical to the ω-zero-sum game except that the defender's goal is to minimize the probability that υ wins. The Stackelberg equilibrium of the υ-zero-sum game (which is equivalent to its Nash equilibria) can be computed in the same way as for the ω-zero-sum game. Let $(\mathbf{x}^\upsilon, a^\upsilon)$ be the equilibrium strategies for the defender and the attacker in the υ-zero-sum game, and let p_0^υ be the probability that υ wins the game given $(\mathbf{x}^\upsilon, a^\upsilon)$. The following proposition shows that $1 - p_0^\upsilon$ is the upper bound of the defender's SSE utility in the game of defense against constructive control.

Proposition 3. *Under the SSE strategies $(\mathbf{x}^*, a^*)$ in the game of defense against constructive control, the defender utility is $u^D(\mathbf{x}^*, a^*) \leq 1 - p_0^\upsilon$.*

Proof. We prove it by contradiction. If $u^D(\mathbf{x}^*, a^*) > 1 - p_0^\upsilon$, it indicates that given $(\mathbf{x}^*, a^*)$, ω succeeds with a probability higher than $1 - p_0^\upsilon$, and thus υ succeeds with probability less than p_0^υ. Since a^* is the attacker's best response to $\mathbf{x}^*$ which maximizes the probability that υ wins, this indicates that $(\mathbf{x}^*, a^*)$ leads to a higher defender utility in the υ-zero-sum game than $(\mathbf{x}^\upsilon, a^\upsilon)$. But in this case, $(\mathbf{x}^\upsilon, a^\upsilon)$ cannot be the equilibrium strategies, which is a contradiction. □

Given Proposition 3, we can further observe that the defender's equilibrium strategy $\mathbf{x}^v$ in the υ-zero-sum game is also her optimal strategy to defend against constructive control whenever the upper bound on the defender's utility is tight. Intuitively, if there exists a pair of equilibrium strategies $(\mathbf{x}^\upsilon, a^\upsilon)$ in the υ-zero-sum game under which either υ or ω wins, i.e., υ wins with a probability of p_0^υ while ω wins with a probability of $1 - p_0^\upsilon$, then $\mathbf{x}^\upsilon$ is the optimal defender strategy to defend against constructive control. Formally, we have the following theorem.

Theorem 5. *If there exists an attacker strategy a so that*

1. *a is the best response to $\mathbf{x}^\upsilon$ in the υ-zero-sum game,*
2. *$P^\omega(\mathbf{x}^\upsilon, a) = 1 - p_0^\upsilon$,*

then (x^v, a) is a pair of SSE strategies in the game of defense against constructive control.

Proof. Since the attacker's goal in the υ-zero-sum game is also to perform constructive control, i.e., to let υ win, a is also the attacker's best response to $\mathbf{x}^{\nu}$ in the game of defense against constructive control. Therefore, based on Proposition 3, (x^{ν}, a) is a pair of SSE strategies in the game. □

The process of computing an approximately optimal strategy for the defender to defend against constructive control then proceeds by solving the two zero-sum games above and choosing the better of the two defender strategies (Algorithm 6). In practice, this approach nearly always finds an optimal solution, as we show in the experiments below.

Algorithm 6: Zero-sum Approximation.

```
$(\mathbf{x}^{\upsilon}, \mathbf{a}^{\upsilon}), p_0^{\upsilon} \leftarrow$ solve $\upsilon$-zero-sum game, $u_1^D = \text{argmax}_{a \in \mathcal{A}^{\upsilon}} P^{\omega}(\mathbf{x}^{\upsilon}, a)$;
/*$\mathbf{a}^{\upsilon}$ is the best mixed strategy of the attacker, $\mathcal{A}^{\upsilon}$ is its support set*/;
if $u_1^D = 1 - p_0^{\upsilon}$ then return $\mathbf{x}^{\upsilon}$;
$(\mathbf{x}^{\omega}, \mathbf{a}^{\omega}) \leftarrow$ solve $\omega$-zero-sum game;
$\mathcal{A}' = \text{argmax}_{a \in \mathcal{A}} P^{\upsilon}(\mathbf{x}^{\omega}, a)$;
/*the set of attacker's best responses to $\mathbf{x}^{\omega}$ to perform constructive control*/;
$u_2^D = \text{argmax}_{a \in \mathcal{A}'} P^{\omega}(\mathbf{x}^{\omega}, a)$;
$u_1^D > u_2^D$ ? return $\mathbf{x}^{\upsilon}$ : return $\mathbf{x}^{\omega}$
```

7. Uncertainty about voter preferences

Our entire treatment of the problem so far assumed complete information about voter preferences for both the attacker and defender. We now show that this assumption is relatively straightforward to relax (from a technical perspective). Specifically, we retain the assumption that the attacker has complete information, but assume that the defender is uncertain about voter preferences. Formally, let V denote a particular voting preference outcome, with R_V being the defender's prior distribution over V. The defender's goal in this setting is to minimize the expected probability that the attacker can successfully control the election. We illustrate defense against destructive control in this section; defense against constructive control can be extended analogously.

Since the attacker knows V, this gives rise to a Bayesian Stackelberg game with V being the attacker's type. Let $W_V^{\omega}(s, a)$ be a binary indicator representing whether ω wins the game given voting preferences V and a strategy pair (s, a). The optimal mixed strategy for the defender can then be computed by solving the following LP, which is a Bayesian extension of the Core-LP in Section 6.1:

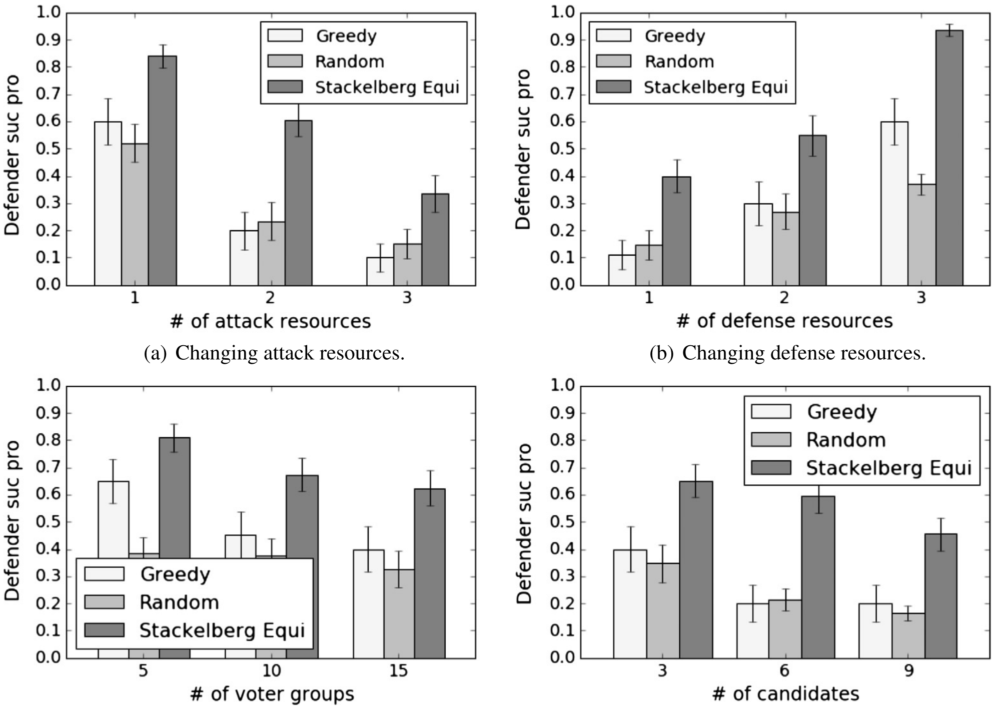
$$\text{Bayesian-LP}(\mathcal{S}, \mathcal{A}): \qquad \max_{\mathbf{x}} \sum_{V} R_V P_V \tag{6a}$$

$$P_V \leq \sum_{s \in \mathcal{S}} x_s W_V^{\omega}(s, a), \quad \forall a \in \mathcal{A}, \forall V \tag{6b}$$

Note that this formulation is amenable to the same double oracle framework that was used to solve the complete information game. The primary difference is that now the attacker oracle must be run for each V, whereas the defender oracle requires a modified objective involving expected probability of the election being controlled with respect to R_V. In practice, since the space of relevant voting preferences V is extremely large, we can take a collection of samples from this distribution and solve the linear program Bayesian-LP solely using these samples to obtain an approximately optimal defense.

8. Evaluation

We evaluate the proposed algorithms in terms of solution quality and scalability. Experiments were run on a Macbook Air computer with 1.7 GHz Intel Core i5, 4 GB memory. Linear and mixed integer programs were solved using CPLEX 12.6.1. We evaluated the algorithms on both synthetic and real data. For synthetic data, we randomly generate a tally for each candidate within each group uniformly in the $[0, 100]$ interval. Each bar in the figures is an average over 101 samples unless otherwise specified, with error bars reflecting a 95% confidence interval. We evaluate solution quality as a function of the following problem parameters: the number of attack resources, the number of defense resources, the number of voter groups, and the number of candidates. Our evaluation on real data uses two datasets. First is the dataset from the 2002 French presidential election [38]. This dataset consists of 2597 votes for 16 candidates by voters in 6 districts (voter groups). The second is the dataset of the 12 largest districts in the state of Michigan in the 2016 U.S. presidential election. In this election, only two of the candidates received enough votes to be meaningfully involved in election control, and consequently constructive and destructive control are equivalent on this dataset.



(a) Changing attack resources.

(b) Changing defense resources.

(c) Changing voter groups.

(d) Changing candidates.

Fig. 1. Solution quality on synthetic data: defense against destructive control.

8.1. Solution quality

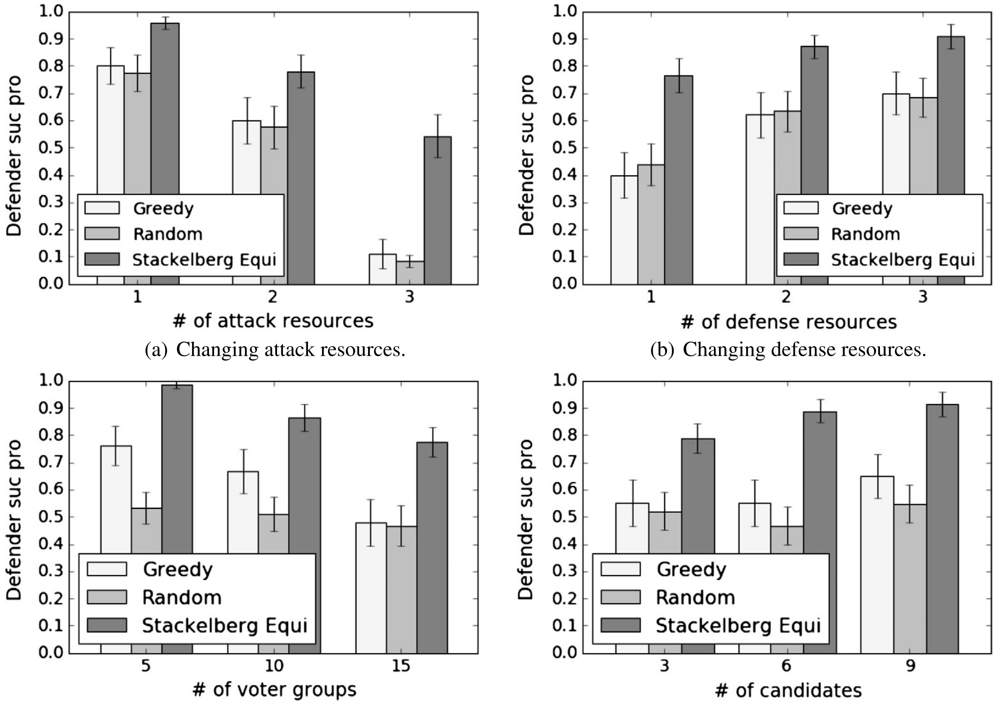
Solution quality of our approach is compared to two baselines. The first, termed *Random*, is a uniformly random defense, that is, the defender plays each pure strategy with the same probability. The second, termed *Greedy*, deterministically protects m groups in which ω has the greatest advantage over the next best candidate in that group. Solution quality is evaluated in terms of the probability that the defender succeeds. More precisely, we aim to maximize the probability that the original winner ω still wins the game under when the attacker plays a best response to the defender's strategy.

Our first evaluation is in the context of defense against destructive control. Figs. 1(a) and 1(b) show the results as a function of the number of attack and defense resources when the number of voter groups is fixed at 10 and the number of candidates is fixed at 4. In Fig. 1(a), the number of defense resources is fixed at 2, while in Fig. 1(b), the number of attack resources is fixed at 2. Figs. 1(c) and 1(d) show the results as a function of the number of voter groups and candidates respectively when the number of resources for both agents is fixed at 2, while in Fig. 1(c) the number of candidates is fixed at 4 and in Fig. 1(d) the number of groups is fixed at 10. 'Stackelberg Equi' refers to the Stackelberg equilibrium solution, i.e., the solution to Core-LP, which always significantly outperforms both baselines in all the experiment settings, in most cases quite dramatically.

Next we perform a similar evaluation in the context of defense against constructive control. For each sample we randomly choose a candidate other than the original winner and assume that the attacker's goal is to make this candidate win. The Stackelberg equilibrium solution is the solution to SSE-MILP. The settings for Figs. 2(a)–2(d) is the same as Figs. 1(a)–1(d). We can readily observe that in this setting as well the Stackelberg equilibrium solution still clearly outperforms the baselines in all test cases.

Having established the effectiveness of the proposed algorithmic approach on synthetic data, we now evaluate it on two real data sets: the real election data from the 2002 French presidential election and the real election data from 2016 American presidential election. We begin with the 2002 French presidential election data and start with defense against destructive control. Figs. 3(a) and 3(b) show the results as a function of the number of attacker/defender resources. The improvement achieved by the proposed algorithms is even more extreme here than when synthetic data is used.

We then evaluate solution quality on the French presidential election data for defense against constructive control. Since there are 16 candidates in total, we assume that the attacker wants each of the 15 candidates who lost the actual election to win in turn. We show the results averaged over these 15 cases as a function of the number of attacker/defender resources in Figs. 4(a) and 4(b) Given the small sample size, the error bars show the standard error rather than 95% confidence intervals in these plots. The Stackelberg equilibrium solution again significantly outperforms the baselines.



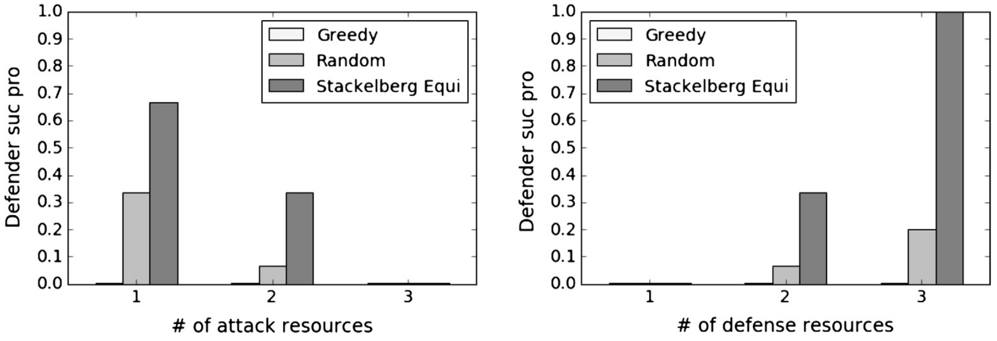
(a) Changing attack resources.

(b) Changing defense resources.

(c) Changing attack resources.

(d) Changing defense resources.

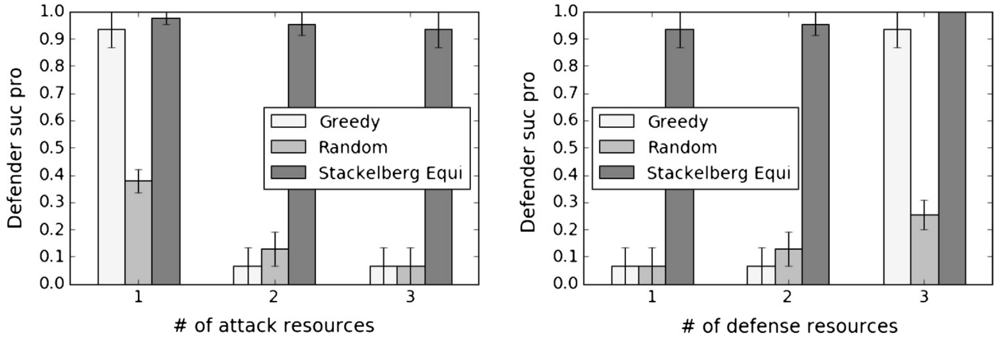
Fig. 2. Solution quality on synthetic data: defense against constructive control.



(a) Changing attack resources.

(b) Changing defense resources.

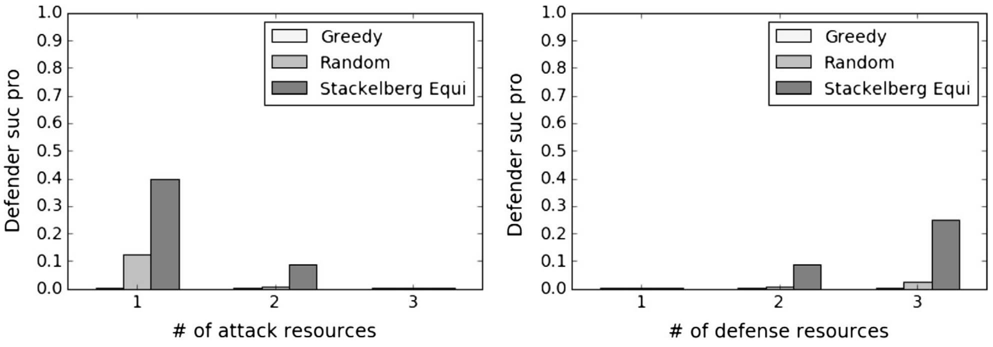
Fig. 3. Solution quality on French Presidential Election data: defense against destructive control.



(a) Changing attack resources.

(b) Changing defense resources.

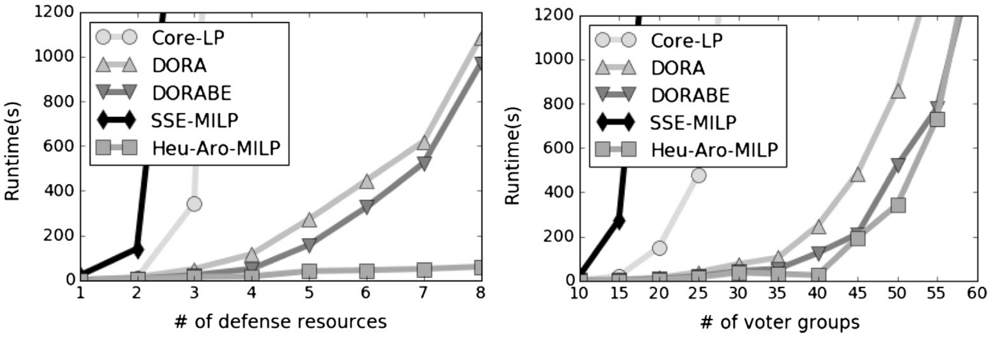
Fig. 4. Solution quality on French Presidential Election data: defense against constructive control.



(a) Changing attack resources.

(b) Changing defense resources.

Fig. 5. Solution quality on American Presidential Election data: defense against destructive control.



(a) Changing defense resources (20 voter groups).

(b) Changing voter groups (3 defense resources).

Fig. 6. Scalability on synthetic data.

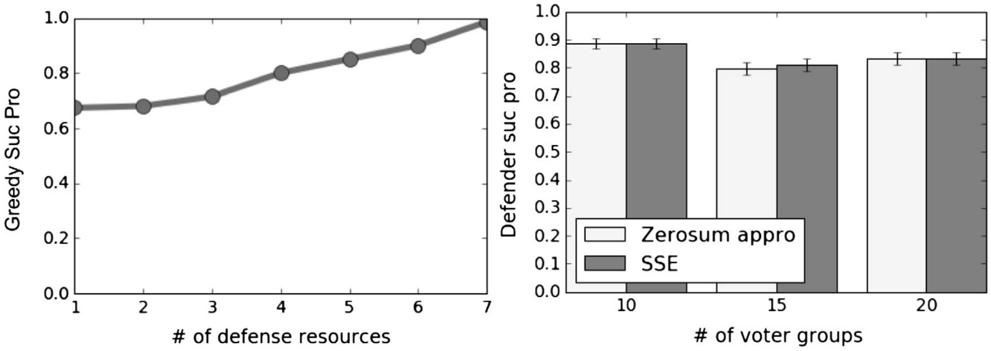
Finally, we evaluate solution quality on the 2016 American presidential election data. Since destructive and constructive control are equivalent for this dataset, we only show the solution quality results for destructive control as a function of the number of attacker/defender resources in Figs. 5(a) and 5(b). As in experiments in previous data sets, the Stackelberg equilibrium solution significantly outperforms the baselines.

8.2. Scalability

Next we compare the scalability of the proposed algorithms. For solving games of defense against destructive control, we evaluate Core-LP algorithm with the two proposed double oracle approaches: 1) using only MILP oracles (DORA), and 2) using the heuristic methods as well (DORABE). For solving games of defense against constructive control, we evaluate SSE-MILP and the heuristic algorithm, which is labeled as 'Heu-Apro-MILP'. In this algorithm, we first run the greedy heuristic algorithm. If it fails to return a defense strategy which can prevent constructive control, we run the zero-sum approximation. If the results of the zero-sum approximation are not optimal (which can be checked based on Theorem 5), then SSE-MILP is called to solve the game.

The results in Fig. 6 show that with increased problem size, either in terms of the number of voter groups or defender resources, the double-oracle approaches significantly outperform Core-LP. We also tested the effect of better oracles. Results show that DORABE usually takes more iterations than DORA to converge, but the runtime of each iteration in DORABE is far less. SSE-MILP is the least scalable, but Heu-Apro-MILP can solve large games quite efficiently. The runtime of Heu-Apro-MILP barely increases as the number of defense resources increases. This is mainly because the greedy heuristic algorithm can directly return the best defense strategy in most cases, especially when there are enough defense resources. As the number of voter groups increases, it shows an increasing trend similar to DORABE.

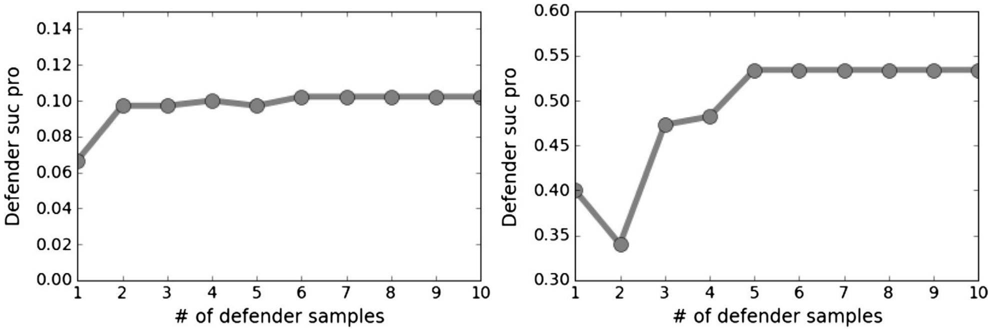
We further evaluate the effect of the greedy heuristic algorithm and the zero sum approximation algorithm for solving games of defense against constructive control. Fig. 7(a) shows the percentage of times the greedy heuristic algorithm succeeds in finding a defender strategy which can prevent constructive control as the number of defense resources increases. Naturally, more resources leads to higher percentage, and it almost always succeeds when there are 7 defense resources (the number of groups is fixed at 20). Fig. 7(b) shows the solution to the zero-sum approximation and the SSE, with each bar group averaged over 500 samples. Results show that solution to the zero-sum approximation is extremely close to the SSE solution in all cases. Indeed, of all 1500 samples, the zero-sum approximation failed to return the SSE solution only once (when the number of groups is 15).



(a) Successful percentage of greedy heuristic.

(b) Zerosum approximation v.s. SSE

Fig. 7. Effect of heuristics.



(a) Low uncertainty.

(b) High uncertainty.

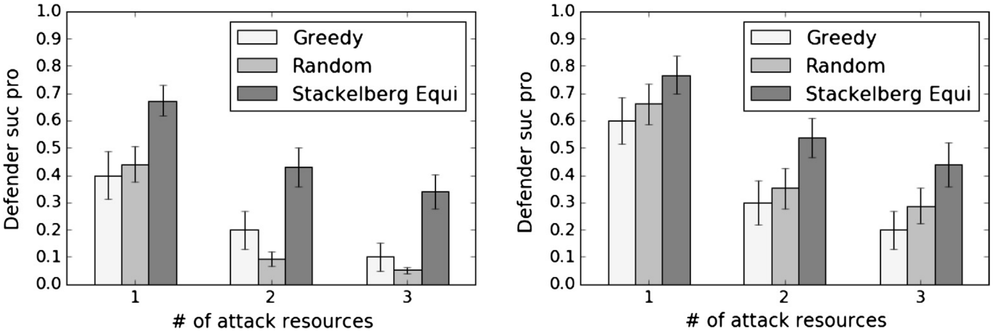
Fig. 8. Bayesian-LP: impact of the number of samples on solution quality.

8.3. Uncertainty

Finally, we compare solution quality of our approach extended to account for defender's uncertainty about voter preferences with the two baselines. The results were qualitatively the same: the Bayesian Stackelberg game approach significantly outperformed the alternatives. We consider the effect of the number of samples from the entire voter preference outcome space used in the Bayesian Stackelberg game to compute an approximate defense under uncertainty. We study two cases: low uncertainty, where the variance of Gaussian noise is 10, and high uncertainty, where tallies of candidates are drawn uniformly in $[1, 400]$ and variance is 20. In both cases, we take 60 attacker types (drawn from this distribution) to be the ground truth. In Figs. 8(a) and 8(b), the x-axis is the number of samples taken by the defender to solve Bayesian-LP, while the y-axis indicates the optimal expected success probability of attackers. We observe that in both experiments very few samples (≤ 6) suffice to achieve a near-optimal solution.

Next, we perform several robustness experiments considering the impact of errors in problem parameters. We first consider that the defender's knowledge about the votes is not accurate, i.e., we sample a collection of tallies as the ground truth, then add a standard Gaussian noise to the votes in each group for each candidate. We assume that the defender strategies are computed based on the votes with noise, the attacker's best response is made based on the original votes, then compute the probability that the defender wins the game given the computed defender strategies, the attacker's best response, and the original votes. Given the defender strategies computed by the proposed algorithms and the baselines, the results in games addressing destructive control and constructive control are shown in Figs. 9(a) and 9(b) respectively. Even with such errors, the proposed algorithms still significantly outperform the baselines in most cases.

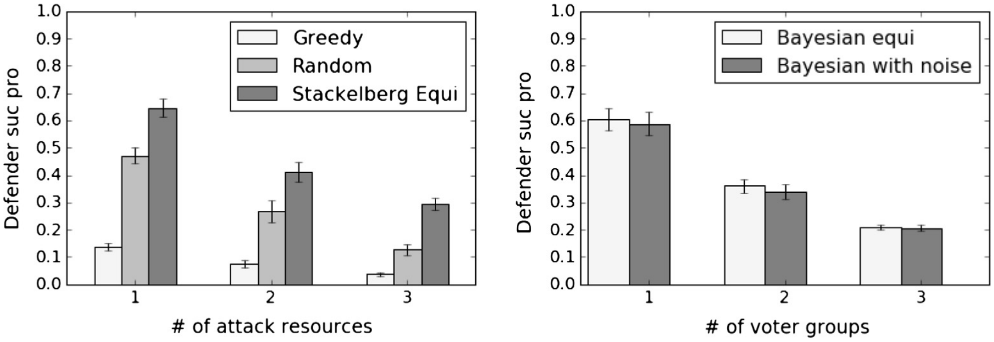
We also consider errors in the defender's knowledge about the probability distribution over types in Bayesian games. We first set a distribution as the ground truth, then add a Gaussian noise with $\mu = 0, \sigma^2 = 0.1$ to the probability that each type appears, then project the new distribution to ensure the sum to be 1. Fig. 10(a) shows the solution quality of the proposed algorithms and the baselines in defense against destructive control. Stackelberg equilibrium still clearly outperforms other solutions. Fig. 10(b) shows the difference between defender's utilities when she computes the Bayesian Stackelberg equilibrium with and without distribution noise, while the attacker's response is always based on the original distribution without noise. Results show that slight errors do not significantly affect solution quality.



(a) Destructive control.

(b) Constructive control.

Fig. 9. Errors in defender's knowledge about votes.



(a) Solution quality.

(b) Original distribution v.s. Distribution with errors.

Fig. 10. Errors in defender's knowledge about vote preference distribution.

9. Conclusion

We study the problem of optimally protecting an election against group-deletion-control, including destructive and constructive control. We show that plurality voting is vulnerable to group-level destructive control, but is resistant to constructive control. In addition, it is computationally hard to protect an election in either destructive or constructive control settings. For defense against destructive control, we propose a double-oracle framework for computing an optimal protection strategy and develop compact mixed integer linear programs for both oracles. We also propose heuristic oracles to further speed up the double oracle framework. For defense against constructive control, we present a mixed integer linear programming approach to compute the SSE of the game. Because this method is itself intractable, we also propose a greedy heuristic algorithm and several zero-sum approximation approaches, which provide upper and lower bounds on the defender's optimal solution. Our experiments demonstrate that the proposed approaches for both destructive and constructive control are effective, far more so than several alternatives. Moreover, we observe that our approach for constructive control which relies on zero-sum approximations actually computes an optimal solution for the defender in nearly every instance.

Acknowledgements

This research was partially supported by the National Science Foundation (CNS-1238959, IIS-1526860, IIS-1649972), Office of Naval Research (N00014-15-1-2621), Army Research Office (W911NF-16-1-0069), AFRL (FA8750-14-2-0180), NRF2015NCR-NCR003-004, and the Israel Science Foundation (grant No. 1488/14).

References

[1] B. An, M. Brown, Y. Vorobeychik, M. Tambe, Security games with surveillance cost and optimal timing of attack execution, in: International Conference on Autonomous Agents and Multiagent Systems, 2013, pp. 223–230.
[2] H. Aziz, I. Schlotter, T. Walsh, Control of fair division, in: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, 2016, pp. 67–73.
[3] J. Bannet, D.W. Price, A. Rudys, J. Singer, D.S. Wallach, Hack-a-Vote: security issues with electronic voting systems, IEEE Secur. Priv. 2 (2004) 32–37.
[4] J.J. Bartholdi, C.A. Tovey, M.A. Trick, How hard is it to control an election? Math. Comput. Model. 16 (1992) 27–40.
[5] D. Baumeister, G. Erdélyi, O.J. Erdélyi, J. Rothe, Computational aspects of manipulation and control in judgment aggregation, in: The Third International Conference on Algorithmic Decision Theory, 2013, pp. 71–85.

[6] N. Betzler, J. Uhlmann, Parameterized complexity of candidate control in elections and related digraph problems, Theor. Comput. Sci. 410 (2009) 5425–5442.
[7] S. Bhattacharjya, Low turnout and invalid votes mark first post war general polls, 2010. http://www.sundaytimes.lk/100411/News/nws_16.html.
[8] F. Brandt, M. Brill, E. Hemaspaandra, L.A. Hemaspaandra, Bypassing combinatorial protections: polynomial-time algorithms for single-peaked electorates, in: AAAI Conference on Artificial Intelligence, 2010, pp. 715–722.
[9] F. Brandt, V. Conitzer, U. Endriss, A.D. Procaccia, J. Lang, Handbook of Computational Social Choice, Cambridge University Press, 2016.
[10] L. Bulteau, J. Chen, P. Faliszewski, R. Niedermeier, N. Talmon, Combinatorial voter control in elections, Theor. Comput. Sci. 589 (2015) 99–120.
[11] J. Chen, P. Faliszewski, R. Niedermeier, N. Talmon, Elections with few voters: candidate control can be easy, in: AAAI Conference on Artificial Intelligence, 2015, pp. 2045–2051.
[12] V. Conitzer, T. Sandholm, Computing the optimal strategy to commit to, in: ACM Conference on Electronic Commerce, 2006, pp. 82–90.
[13] Y. Desmedt, E. Elkind, Equilibria of plurality voting with abstentions, in: Proceedings of the 11th ACM Conference on Electronic Commerce, 2010, pp. 347–356.
[14] E. Elkind, H. Lipmaa, Hybrid voting protocols and hardness of manipulation, in: International Symposium on Algorithms and Computation, 2005, pp. 206–215.
[15] G. Erdélyi, M.R. Fellows, J. Rothe, L. Schend, Control complexity in Bucklin and fallback voting: a theoretical analysis, J. Comput. Syst. Sci. 81 (2015) 632–660.
[16] G. Erdélyi, M.R. Fellows, J. Rothe, L. Schend, Control complexity in Bucklin and fallback voting: an experimental analysis, J. Comput. Syst. Sci. 81 (2015) 661–670.
[17] G. Erdélyi, E. Hemaspaandra, L.A. Hemaspaandra, Bribery and voter control under voting-rule uncertainty, in: International Conference on Autonomous Agents and Multi-Agent Systems, 2014, pp. 61–68.
[18] G. Erdélyi, E. Hemaspaandra, L.A. Hemaspaandra, More natural models of electoral control by partition, in: International Conference on Algorithmic Decision Theory, 2015, pp. 396–413.
[19] G. Erdélyi, M. Nowak, J. Rothe, Sincere-strategy preference-based approval voting fully resists constructive control and broadly resists destructive control, Math. Log. Q. 55 (2009) 425–443.
[20] G. Erdélyi, C. Reger, Y. Yang, The complexity of bribery and control in group identification, in: Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, 2017, pp. 1142–1150.
[21] P. Faliszewski, E. Hemaspaandra, L.A. Hemaspaandra, Multimode control attacks on elections, J. Artif. Intell. Res. 40 (2011) 305–351.
[22] P. Faliszewski, E. Hemaspaandra, L.A. Hemaspaandra, Weighted electoral control, J. Artif. Intell. Res. 52 (2015) 507–542.
[23] P. Faliszewski, E. Hemaspaandra, L.A. Hemaspaandra, J. Rothe, Llull and Copeland voting computationally resist bribery and constructive control, J. Artif. Intell. Res. 35 (2009) 275–341.
[24] P. Faliszewski, E. Hemaspaandra, L.A. Hemaspaandra, J. Rothe, The shield that never was: societies with single-peaked preferences are more open to manipulation and control, Inf. Comput. 209 (2011) 89–107.
[25] Z. Fitzsimmons, E. Hemaspaandra, L.A. Hemaspaandra, Control in the presence of manipulators: cooperative and competitive cases, in: International Joint Conference on Artificial Intelligence, 2013, pp. 113–119.
[26] J. Gan, B. An, Y. Vorobeychik, Security games with protection externality, in: AAAI Conference on Artificial Intelligence, 2015, pp. 914–920.
[27] E. Hemaspaandra, L.A. Hemaspaandra, J. Rothe, Anyone but him: the complexity of precluding an alternative, Artif. Intell. 171 (2007) 255–285.
[28] E. Hemaspaandra, L.A. Hemaspaandra, J. Rothe, Hybrid elections broaden complexity-theoretic resistance to control, Math. Log. Q. 55 (2009) 397–424.
[29] E. Hemaspaandra, L.A. Hemaspaandra, J. Rothe, The complexity of controlling candidate-sequential elections, Theor. Comput. Sci. 678 (2017) 14–21.
[30] E. Hemaspaandra, L.A. Hemaspaandra, J. Rothe, The complexity of online voter control in sequential elections, Auton. Agents Multi-Agent Syst. 31 (2017) 1055–1076.
[31] E. Hemaspaandra, L.A. Hemaspaandra, H. Schnoor, A control dichotomy for pure scoring rules, in: AAAI Conference on Artificial Intelligence, 2014, pp. 712–720.
[32] L.A. Hemaspaandra, R. Lavaee, C. Menton, Schulze and ranked-pairs voting are fixed-parameter tractable to bribe, manipulate, and control, Ann. Math. Artif. Intell. 77 (2016) 191–223.
[33] A. Horn, US recounts find no evidence of hacking in Trump win but reveal vulnerabilities, Guardian, 2016. https://www.theguardian.com/us-news/2016/dec/28/election-recount-hacking-voting-machines (Accessed 19 October 2017).
[34] M. Jain, V. Conitzer, M. Tambe, Security scheduling for real-world networks, in: International Conference on Autonomous Agents and Multi-Agent Systems, 2013, pp. 215–222.
[35] C. Kiekintveld, M. Jain, J. Tsai, J. Pita, F. Ordóñez, M. Tambe, Computing optimal randomized resource allocations for massive security games, in: International Joint Conference on Autonomous Agents and Multiagent Systems, 2009, pp. 689–696.
[36] D. Korzhyk, V. Conitzer, R. Parr, Security games with multiple attacker resources, in: International Joint Conference on Artificial Intelligence, 2011, pp. 273–279.
[37] D. Korzhyk, Z. Yin, C. Kiekintveld, V. Conitzer, M. Tambe, Stackelberg vs. Nash in security games: an extended investigation of interchangeability, equivalence, and uniqueness, J. Artif. Intell. Res. 41 (2011) 297–327.
[38] J.F. Laslier, K. Van der Straeten, A live experiment on approval voting, Exp. Econ. 11 (2008) 97–105.
[39] H. Liu, H. Feng, D. Zhu, J. Luan, Parameterized computational complexity of control problems in voting systems, Theor. Comput. Sci. 410 (2009) 2746–2753.
[40] H. Liu, D. Zhu, Parameterized complexity of control problems in maximin election, Inf. Process. Lett. 110 (2010) 383–388.
[41] A. Loreggia, N. Narodytska, F. Rossi, K.B. Venable, T. Walsh, Controlling elections by replacing candidates or votes, in: International Conference on Autonomous Agents and Multiagent Systems, 2015, pp. 1737–1738.
[42] K. Magiera, P. Faliszewski, How hard is control in single-crossing elections? Auton. Agents Multi-Agent Syst. 31 (2017) 606–627.
[43] N. Mattei, N. Narodytska, T. Walsh, How hard is it to control an election by breaking ties?, in: Proceedings of the Twenty-First European Conference on Artificial Intelligence, 2014, pp. 1067–1068.
[44] C. Maushagen, J. Rothe, Complexity of control by partitioning veto and maximin elections and of control by adding candidates to plurality elections, in: European Conference on AI, 2016, pp. 277–285.
[45] C. Maushagen, J. Rothe, Complexity of control by partition of voters and of voter groups in veto and other scoring protocols, in: Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, 2017, pp. 615–623.
[46] G.P. McCormick, Computability of global solutions to factorable nonconvex programs: part I—convex underestimating problems, Math. Program. 10 (1976) 147–175.
[47] H.B. Mcmahan, G.J. Gordon, A. Blum, Planning in the presence of cost functions controlled by an adversary, in: International Conference on Machine Learning, 2003, pp. 536–543.
[48] C. Menton, Normalized range voting broadly resists control, Theory Comput. Syst. 53 (2013) 507–531.
[49] T. Miasko, P. Faliszewski, The complexity of priced control in elections, Ann. Math. Artif. Intell. 77 (2016) 225–250.
[50] M. Neveling, J. Rothe, Solving seven open problems of offline and online control in Borda elections, in: AAAI Conference on Artificial Intelligence, 2017, pp. 3029–3035.

[51] D. Parkes, L. Xia, A complexity-of-strategic-behavior comparison between Schulze's rule and ranked pairs, in: AAAI Conference on Artificial Intelligence, 2012, pp. 1429–1435.

[52] P. Paruchuri, J.P. Pearce, J. Marecki, M. Tambe, F. Ordonez, S. Kraus, Playing games for security: an efficient exact algorithm for solving Bayesian Stackelberg games, in: International Joint Conference on Autonomous Agents and Multiagent Systems, 2008, pp. 895–902.

[53] T. Put, P. Faliszewski, The complexity of voter control and shift bribery under parliament choosing rules, in: Transactions on Computational Collective Intelligence XXIII, Springer, 2016, pp. 29–50.

[54] A. Rey, J. Rothe, Structural control in weighted voting games, in: 41st International Symposium on Mathematical Foundations of Computer Science, MFCS, 2016, pp. 80:1–80:15.

[55] J. Rothe, L. Schend, Challenges to complexity shields that are supposed to protect elections against manipulation and control: a survey, Ann. Math. Artif. Intell. 68 (2013) 161–193.

[56] M. Rubinkam, F. Bajak, Analysis: recounts or no, U.S. elections are still vulnerable to hacking, Chicago Tribune, 2016. http://www.chicagotribune.com/news/nationworld/ct-election-hacking-recount-20161226-story.html (Accessed 19 October 2017).

[57] Agencies Staff, Pakistani elections hit by bomb attacks, Guardian, 2013. https://www.theguardian.com/world/2013/may/11/pakistani-elections-bomb-blast-karachi (Accessed 19 October 2017).

[58] M. Tambe, Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned, Cambridge University Press, 2011.

[59] Y. Vorobeychik, J. Letchford, Securing interdependent assets, Auton. Agents Multi-Agent Syst. 29 (2015) 305–333.

[60] Z. Wang, Y. Yin, B. An, Computing optimal monitoring strategy for detecting terrorist plots, in: AAAI Conference on Artificial Intelligence, 2016, pp. 637–643.

[61] E. Watkins, Cheney: Putin made 'a very serious effort' to interfere in US election, CNN, 2017. http://www.cnn.com/2017/03/27/politics/dick-cheney-russia/index.html (Accessed 19 October 2017).

[62] S. Wolchok, E. Wustrow, D. Isabel, J.A. Halderman, Attacking the Washington, DC internet voting system, in: International Conference on Financial Cryptography and Data Security, 2012, pp. 114–128.

[63] L. Xia, V. Conitzer, Stackelberg voting games: computational aspects and paradoxes, in: AAAI, 2010, pp. 805–810.

[64] L. Xu, F. Hutter, H.H. Hoos, K. Leyton-Brown, SATzilla: portfolio-based algorithm selection for SAT, J. Artif. Intell. Res. 32 (2008) 565–606.

[65] Y. Yang, On the complexity of Borda control in single-peaked elections, in: Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, 2017, pp. 1178–1186.

[66] Y. Yang, J. Guo, How hard is control in multi-peaked elections: a parameterized study, in: International Conference on Autonomous Agents and Multi-agent Systems, 2015, pp. 1729–1730.

[67] Y. Yang, J. Wang, Anyone but them: the complexity challenge for a resolute election controller, in: Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems, 2017, pp. 1133–1141.

[68] Y. Yin, Y. Vorobeychik, B. An, N. Hazon, Optimally protecting elections, in: International Joint Conference on Artificial Intelligence, 2016, pp. 681–687.

[69] M. Zuckerman, P. Faliszewski, Y. Bachrach, E. Elkind, Manipulating the quota in weighted voting games, Artif. Intell. 180–181 (2012) 1–19.