Optimally Protecting Elections

Yue Yin[1,2], **Yevgeniy Vorobeychik**[3], **Bo An**[4], **Noam Hazon**[5]
[1]Key Lab of Intelligent Information Processing, ICT, CAS, [2]University of CAS, Beijing, China
[3]Dept. of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN
[4]School of Computer Science and Engineering, Nanyang Technological University, Singapore
[5]Dept. of Computer Science, Ariel University, Israel
[1]yiny@ics.ict.ac.cn, [3]yevgeniy.vorobeychik@vanderbilt.edu, [4]boan@ntu.edu.sg, [5]noamh@ariel.ac.il

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 protection against election 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 show that for plurality voting, election control through group deletion to prevent a candidate from winning is in P, while it is NP-Hard to prevent such control. We then present a double-oracle framework for computing an optimal prevention strategy, developing exact mixed-integer linear programming formulations for both the defender and attacker oracles (both of these subproblems we show to be NP-Hard), as well as heuristic oracles. Experiments conducted on both synthetic and real data demonstrate that the proposed computational framework can scale to realistic problem instances.

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 to their own goals. Indeed, actual incidents of vote manipulation and control, sometimes through violence, bear out this concern. For example, the 2013 election in Pakistan was marred by a series of election-day bombings, resulting in over 30 dead and 200 injured, in an attempt to subvert the voting process [RT, 2013], and the 2010 Sri Lanka election exhibited 84 major and 202 minor incidents of poll-related violence [Bhattacharjya, 2010]. Moreover, with the dawn of electronic and Internet voting, the additional threat of election control and manipulation through cyber means has emerged, with a number of documented demonstration attacks [Bannet *et al.*, 2004; Wolchok *et al.*, 2012].

The study of the computational complexity of election control was initiated by Bartholdi *et al.* [1992] as a novel defense against election control. 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 [Bartholdi *et al.*, 1992; Hemaspaandra *et al.*, 2007]. 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 [Erdélyi *et al.*, 2009; Hemaspaandra *et al.*, 2009]. 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., Xu *et al.* [2008] 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 a specific candidate from winning. We show that for plurality voting optimal election control in this model can be computed in polynomial time. Next, we consider the problem of protection against election 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. Protection resources may represent actual physical security for polling centers or voting stations, or resources devoted to frequent auditing of spe-

cific electronic voting systems. In this model, 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 that guarantee that an election cannot be controlled is NP-Hard. For the more general problem, we propose a double-oracle framework to compute an optimal protection. We prove that both the defender, and attacker oracles are NP-Hard 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. Moreover, we develop heuristic defender and attacker oracles which significantly speed up the framework. Our experiments demonstrate the effectiveness and scalability of our algorithmic approach.

In summary, we make the following contributions:

- A new model of protecting elections from group-level election control attacks,
- A polynomial-time algorithm for group-level election control,
- Complexity analysis of guaranteeing that an election cannot be controlled,
- A scalable double-oracle framework for choosing optimal allocation of protection resources.

2 Related Work

The study of the computational complexity of election control was initiated by Bartholdi *et al.* [1992], who analyzed plurality and Condorcet voting with several types of control. While Bartholdi *et al.* studied the constructive variant of the control problem, where the goal is to ensure a given candidate's victory, we study a 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.* [2007], who also analyzed the approval voting rule. The study of election control was further extended to a number of other models and voting rules [Betzler and Uhlmann, 2009; Liu *et al.*, 2009; Liu and Zhu, 2010; Faliszewski *et al.*, 2011; Parkes and Xia, 2012; Faliszewski *et al.*, 2013; Menton, 2013]. However, all of these consider the election control problem at the granularity of individual voters. The work of Chen *et al.* [2014] was 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. That is, the voters are grouped according to their preferences and the groups can overlap. In our setting the voters are grouped arbitrarily, with no overlap, and the election control is by deleting (groups of) voters. Recently, Erdélyi *et al.* [2015] studied election control of plurality by adding or deleting groups of voters, but they only consider the variant of constructive control. Finally, Chen *et al.* [2015] studied constructive and destructive control by adding or deleting groups of *candidates* (but not voters).

There has been extensive research on modeling physical security problems using Stackelberg games [Tambe, 2011]. Much of prior work has focused on attackers who can only attack a single target [Gan *et al.*, 2015; An *et al.*, 2013]. Exceptions to this involved either simultaneous-move scenarios [Korzhyk *et al.*, 2011a] or heuristic approaches [Vorobeychik and Letchford, 2015]. 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 [Mcmahan *et al.*, 2003; Jain *et al.*, 2013; Wang *et al.*, 2016]. However, as oracles are model dependent, the special structure of our problem requires novel scalable algorithms.

3 Election Control by Deleting Voter Groups

A common question in election control is whether it is possible to prevent a specific candidate from winning by deleting a subset of voters. We begin by analyzing this destructive control problem whereby we allow attackers to delete (or deploy a denial-of-service attack against) groups of voters, which may represent polling locations. 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. Throughout, we focus 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). For each group $i \in I$, let v_{ic} be the number of votes for candidate c, and let $v_c = \sum_i v_{ic}$ 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: $\omega = \arg\max_c v_c$. We now consider the problem of election control in which the attacker may choose to delete a subset of at most $k \leq n$ groups, with the goal of preventing ω from winning.[1]

It is well known that optimal constructive and destructive control of plurality by deleting individual voters can be computed in polynomial time [Bartholdi *et al.*, 1992; Hemaspaandra *et al.*, 2007]. Allowing the attacker to select specific groups may appear to significantly complicate the problem. Indeed, [Erdélyi *et al.*, 2015] showed that this type of constructive control is NP-Complete even with plurality. Surprisingly, we show that the destructive variant can still be computed in polynomial time, significantly generalizing the previous result of [Hemaspaandra *et al.*, 2007]. 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 $d^c = \langle d_i^c : i \in I \rangle$ be a vector with $d_i^c = v_{ic} - v_{i\omega}$, that is, the vote advantage of c over ω in group $i \in I$. For a vector d^c, define $sum(d^c) = \sum_i d_i^c$. Then, $sum(d^c)$ is the total difference of votes between c and ω. For

[1] Note that "traditional" election control by deleting votes is a special case of our setting, where each group contains a single voter.

example, suppose that d^c 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 d_i^c to verify whether it is possible to make c have a larger vote count than ω. For convenience, define d^{c-k} to be the portion of the vector d^c remaining after the k groups with smallest d_i^c have been deleted.

Proposition 1. *For a given candidate $c \in C$, it is possible to delete k groups to ensure that $v_c > v_\omega$ iff $sum(d^{c-k}) > 0$.*

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

The process of computing a group-level election control approach is shown in Algorithm 1. For each candidate $c \in C \backslash \{\omega\}$, 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. It is not difficult to see that the complexity of Algorithm 1 is O($|C|n \log n$), which yields the following:

Algorithm 1: Optimal Election Control by Group Deletion

```
1 for c ∈ C^{-ω} do
2     d^{c-k} ← Sort d^c in ascending order, delete the first k
        elements in d^c;
3     if sum(d^{c-k}) > 0 then
4         return Attack voter groups corresponding to deleted
            elements;
5 return No control approach;
```

Theorem 1. *Election control preventing a candidate ω from winning by deleting k voter groups can be accomplished in O($|C|n \log n$) time.*

4 Protecting Elections

Given that plurality is extremely vulnerable to control by deleting voter groups, we now pose the dual question: is it possible for a party interested in maintaining election integrity (henceforth, *defender*) to ensure that plurality is resilient to control? To address this question, we consider the following model of protection. The defender can deploy $m \leq n$ protection resources (e.g., physical protection, electronic auditing, etc) to protect individual voter groups from attacks. If a group i is protected, we assume that it cannot be deleted by the adversary. We now ask: how hard is it for the defender to guarantee that a given set of resources m is sufficient to protect the election, that is, to ensure that it is impossible for an attacker to make ω lose by deleting unprotected voter groups?

Definition 1 (Hitting Set Problem)**.** *A set G, 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 is sufficient to prevent control is NP-Complete.*

Proof. It is easy to see that this decision problem is in NP. To show that it is NP-Hard, we reduce from the hitting set problem. Specifically, we show that for any hitting set problem, we can construct an election with n voter groups, so that there exists a hitting set G' iff 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 a 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 we assume that $d_i^c = -1$, i.e., c has 1 fewer vote than ω in group i; otherwise let $d_i^c = 0$, i.e., c and ω ties in group i. Assume that up to $k = n - m$ groups are attacked.

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 $d_i^c = -1$ is protected, i.e., $G' \cap \hat{G} \neq \emptyset$. This is because if there exists a candidate c, all voter groups with $d_i^c = -1$ are unprotected, then the attacker can successfully attack all such groups and c will tie with ω in the remaining votes. Thus the protected voter groups satisfy that $\forall \hat{G} \in U, G' \cap \hat{G} \neq \emptyset$, which is a 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. □

Theorem 2 leaves us with two questions: 1) does this mean that we cannot protect elections in practice, and 2) is all hope lost if m is insufficient to protect an election? In answering question 2, clearly we cannot protect the election if protection resources are allocated deterministically. However, when resources are limited, randomized allocation can offer tremendous value, increasing uncertainty and raising the stakes for attackers [Paruchuri *et al.*, 2008]. We propose to address both of these questions through a single framework: a Stackelberg game model in which the defender (of the election) first chooses a randomized allocation of m protection resources, and the attacker follows by choosing k groups to attack. Formally, let s denote a pure strategy of the defender, where $s_i \in \{0, 1\}$ indicates whether a voter group i is protected. Similarly, the attacker's pure strategy is a vector a where a_i indicates whether group i is attacked. We use $\mathcal{S}$ and $\mathcal{A}$ to represent the strategy space of the defender and the attacker

respectively. Let $P(s,a) \in \{0,1\}$ be an indicator denoting whether an attack a succeeds when a pure protection strategy s is played. Implicitly, we have assumed that both the attacker and defender know the net voting tallies for each location i. We relax this assumption in Section 6. Utilities of the attacker and defender are then defined by $u_A(s,a) = P(s,a)$ and $u_D(s,a) = -P(s,a)$, respectively, so that the game is zero-sum. Since we allow randomization for the defender, let $\mathbf{x}$ denote its randomized (mixed) strategy, with x_s the probability that a pure strategy $s \in \mathcal{S}$ is used.

Since the game is zero-sum, the Stackelberg equilibrium strategy for the defender is equivalent to its Nash equilibria [Korzhyk *et al.*, 2011b]. Consequently, one can use a well-known linear programming formulation, shown as a Linear Program 1b (henceforth, Core-LP) below, for solving zero-sum normal-form games [Conitzer and Sandholm, 2006].

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

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

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 cases the strategy space for both players is combinatorial, this is a non-starter. We therefore develop a *Double Oracle* approach for addressing this scalability issue.

5 Double Oracle Approach

The double oracle framework is a common approach for solving zero-sum games with exponential strategy spaces of both players [Mcmahan *et al.*, 2003; Jain *et al.*, 2013]. 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 re-solved. 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 dependent. For example, none of the prior approaches (e.g., [Jain *et al.*, 2013]) 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: S' ⊂ S; A' ⊂ A;
2  while do
3      (x, y) ← Core-LP(S', A');
4      a ← AO-Better(x);
5      if a = ∅ then a ← AO-MILP(x);
6      s ← DO-Better(y);
7      if s = ∅ then s ← DO-MILP(y);
8      if a ∈ A and s ∈ S then
9          return x;
10     else
11         A' ← A' ∪ {a}, S' ← S' ∪ {s};
```

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

5.1 Attacker Oracle

Complexity: It would seem that in Theorem 1 we had already shown that controling election in our model is in P. However, this result assumed that no protection is deployed (equivalently, that protection is deterministic). Surprisingly, when protection is randomized, election control, which we also refer to as the attacker's *best response* or *oracle*, is NP-Hard, as the following result attests (in this result, $\mathcal{S}'$ represents the support of the defender's mixed strategy).

Theorem 3. *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. It is easy to see that this decision problem is in NP. To show that it is NP-Hard, we reduce from the hitting set problem shown in Definition 1. Specifically, we show that for any 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' iff there exists an attacker strategy which can control the election no matter which defender strategy $s \in \mathcal{S}'$ is played.

Given a hitting set problem as is shown in Definition 1, we construct an election with $|G|+1$ voter groups, two candidates, ω and another candidate c. Each $i \in G$ corresponds to a voter group, in which we assume that $d_i^c = -1$, 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 $d_j^c = |G| - 1$. 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 $U = \{\{1,2\}\}$. Then there are 4 groups. In the first three 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 $\{1,2\} \in U$, group 3 and 4 are protected.

The $\Leftarrow$ direction: If there exists an attacker strategy which attacks k 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 $d_i^c = -1$ is attacked. Thus $\forall \hat{G} \in U, G' \cap \hat{G} \neq \emptyset$. Otherwise the attacker cannot control the election if s is played. Therefore, G' is a required hitting set.

The $\Rightarrow$ direction: Given a hitting set $G' \subset G$ with $|G'| = k$, 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 $d_i^c = -1$ 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. □

Exact Solution: Although computing attacker's best response (oracle) is NP-Hard, 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 solving $\max_{a \in \mathcal{A}} \sum_{s \in \mathcal{S}'} P(s,a) x_s$ for a given mixed strategy $\mathbf{x}$. Our first step is to formulate the attacker oracle as a mathematical (non-linear) program. The main technical challenge involved is representing $P(s,a)$, which is a non-trivial function of s and a. We do this implicitly in AO-MP by using an auxiliary binary variable z_s.

$$\text{AO-MP}: \quad \max_{a_i, z_s, e_s^c \in \{0,1\}} \quad \sum\nolimits_{s \in \mathcal{S}'} \quad z_s \cdot x_s \tag{2a}$$

$$\sum\nolimits_i a_i \leq k \tag{2b}$$

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

$$z_s \sum\nolimits_{c \in C^{-\omega}} e_s^c \left(\sum\nolimits_i d_i^c (1 - (1 - s_i) a_i) \right) \geq 0, \forall s \in \mathcal{S}'. \tag{2d}$$

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 $d^{c\prime} = \sum_i d_i^c - \sum_i (1 - s_i) a_i d_i^c$. Note that $z_s = 1$, i.e., attacker succeeds 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., $d^{c\prime} \geq 0$. 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, $\sum_{c \in C^{-\omega}} e_s^c \left(\sum_i d_i^c (1 - (1 - s_i) a_i) \right) \geq 0$, and the associated $z_s = 1$, yielding, in combination with Constraint (2b) a pure strategy for the attacker that maximizes its success probability 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 [McCormick, 1976], yielding an MILP for computing the attacker's best response.

Heuristic "Better" Response: The main issue with AO-MP 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 lose no solution guarantee.

We take two steps to find a better response for the attacker. First, we look for a subset $\mathcal{S}'' \subset \mathcal{S}'$ with $\sum_{s \in \mathcal{S}''} x_s > p$, where p is the objective value of Core-LP restricted to a small subset of attacker strategies $\mathcal{A}'$ in the previous iteration. Second, we look for an attacker pure strategy a which can successfully affect the voting result no matter which pure strategy $s \in \mathcal{S}''$ is played by the defender, i.e., $P(s,a) = 1 \; \forall s \in \mathcal{S}''$. If we can successfully find such a set $\mathcal{S}''$ and a pure strategy a, the attacker will succeed with a probability of at least $\sum_{s \in \mathcal{S}''} x_s$ if he plays pure strategy a. Since $\sum_{s \in \mathcal{S}''} x_s > p$, 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 to success 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: S', x, p;
2  S̄ = ⟨s^ρ, ρ ∈ 1,2,3,···⟩ ← sort s ∈ S' by decreasing x_s;
3  for ρ in 1..|S̄| do
4      p' ← x_{s^ρ}, S'' ← {s^ρ}, ρ' ← ρ + 1;
5      while p' ≤ p and ρ' ≤ |S̄| do
6          p' ← p' + x_{s^ρ'}, S'' ← S'' ∪ {s^ρ'}, ρ' ← ρ' + 1;
7      if p' > p then
8          for c ∈ C^{-ω} do
9              d^{c'} ← ⟨d_i^c : i with s_i = 0, ∀s ∈ S''⟩;
10             d^{(c-k)'} ← delete the smallest k elements in d^{c'};
11             if sum(d^{(c-k)'}) ≥ 0 then
12                 return attack the k groups corresponding to
                   deleted elements;
13 return ∅;
```

5.2 Defender Oracle

We now proceed to analyze the NP-Hard defender oracle (Theorem 2).

Exact Solution: The defender's oracle, or best response, can be defined as: $\max_{s \in \mathcal{S}} \sum_{a \in \mathcal{A}'} (1 - P(s,a)) y_a$. Just as in the attacker oracle formulation, we proceed to develop the (non-linear) mathematical integer program to compute the defender's best response.

$$\text{DO-MP:} \qquad \max_{s_i, z_a \in \{0,1\}} \quad \sum_{a \in \mathcal{A}'} z_a \cdot y_a \tag{3a}$$

$$\sum\nolimits_i s_i \leq m \tag{3b}$$

$$z_a \sum\nolimits_i (d_i^c (1 - (1 - s_i) a_i) + 1) \leq 0, \forall c \in C^{-\omega}, a \in \mathcal{A}'. \tag{3c}$$

There is an important difference from the attacker oracle: in particular, $z_a = 1$ (that is, the defender successfully blocks an attack strategy $a \in \mathcal{A}'$, 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 (3c) ensures that $z_a = 1$ only when $\forall c \in C^{-\omega}$, $\sum_i d_i^c - \sum_i (1 - s_i) a_i d_i^c < 0$, while Constraint (3b) enforces feasibility of the defender's strategy. The resulting DO-MP thereby chooses the defender strategy which minimizes the probability of a successful attack for a fixed attacker mixed strategy $\mathbf{y}$. We can then linearize the nonlinear constraint (3c) by using McCormick inequalities [McCormick, 1976], obtaining an MILP formulation of the defender oracle.

Heuristic "Better" Response (Algorithm 4): We first look for a subset $\mathcal{A}'' \subset \mathcal{A}'$ with $\sum_{a \in \mathcal{A}''} y_a > 1 - p$. Then we look for a defender pure strategy s which can "block" all attacker strategies $a \in \mathcal{A}''$, ensuring that the attacker will succeed with probability less 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 = ⟨s_i = 0 : ∀i ∈ {1, ··· , n}⟩, res = 0;
for each A'' with Σ_{a∈A''} y_a > 1 − p do
    for c ∈ C^{−ω} do
        d^{c'} ← ⟨d_i^c : i with a_i = 0, ∀a ∈ A'' or s_i = 1⟩;
        while sum(d^{c'}) ≥ 0 and res < m do
            d^{c''} ← d^c \ d^{c'}, i* ← argmin_i{d_i^{c''}};
            d^{c'} ← d^{c'} ∪ {d_{i*}^c}, s_{i*} ← 1, res ← res + 1;
    if ∀c ∈ C^{−ω}, d^{c'} < 0 then
        return s;
return ∅;
```

6 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 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. Since the attacker knows V, this gives rise to a Bayesian Stackelberg game with V the attacker's type. Let $p_V(s, a)$ be a binary indicator representing whether the attacker can successfully control the voting 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 above:

$$\text{Bayesian-LP}(\mathcal{S}, \mathcal{A}): \qquad \min_{\mathbf{x}} \sum_V R_V P_V \tag{4a}$$

$$P_V \geq \sum_{s \in \mathcal{S}} x_s p_V(s, a), \quad \forall a \in \mathcal{A}, \forall V \tag{4b}$$

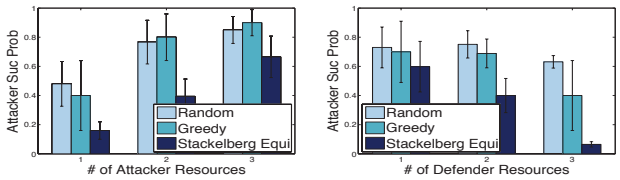
Note that this formulation is amenable to the same double oracle framework that was used to solve the complete information game. The primary difference is that now the attacker oracle must be run for each V, whereas the defender oracle requires a modified objective involving expected probability of the 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 (4) solely using these samples to obtain an approximately optimal defense.

7 Evaluation

We evaluate the proposed algorithms on both synthetic and real data with respect to solution quality and scalability. Solution quality of our approach is compared to two baselines. The first, termed *Random*, is a uniformly random defense. The second, termed *Greedy*, deterministically protects m groups in which ω has the greatest advantage over the next best candidate in that group. Linear and mixed integer programs were solved using CPLEX 12.6.1.We randomly generated a tally for each candidate within each group uniformly in $[0, 100]$. Each data point is an average over 30 such samples.

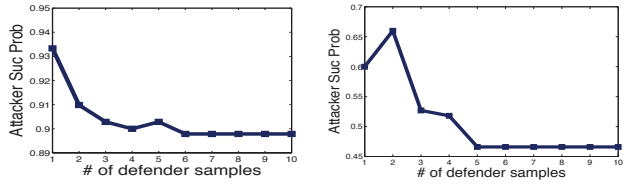
Figures 1(a) and 1(b) show the solution quality of the proposed algorithms and the baselines when there are 30 voter groups and 5 candidates. The Stackelberg equilibrium solution always outperforms both baselines above, in most cases quite dramatically. We also tested the algorithms on other combinations of voters and candidates and observed similar results. In addition, we compared 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. In addition, 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 model uncertainty by taking baseline voting tallies (generated as described above), and adding zero-mean Gaussian noise. 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 Figures 2(a) and 2(b), the x-axis is the number of samples taken by the defender to solve Baysian-LP, while the y-axis indicates the optimal expected success probability of attackers. We observe that in both treatments very few samples (≤ 6) suffice to achieve a near-optimal solution. Additionally, we performed several robustness experiments, considering the impact of errors in problem parameters (e.g., voter preferences, in the complete information case and probability distribution over types in Bayesian games) on solution quality. We found that solutions are robust to such errors.

Next we compare the scalability of the Core-LP algorithm with the two proposed double oracle approaches: 1) using only MILP oracles (DORA), and 2) using the heuristic meth-



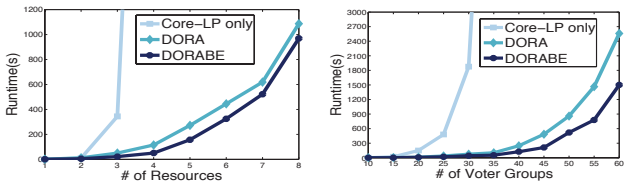
(a) Changing attack resources. (b) Changing defense resources.

Figure 1: Comparison of solution quality on synthetic data. "Stackelberg Equi" is the Stackelberg equilibrium solution.



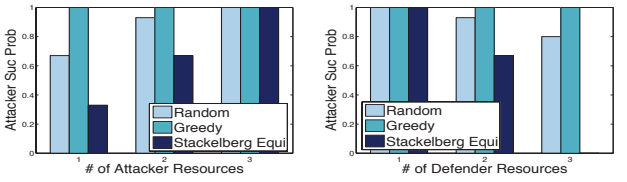
(a) Low uncertainty. (b) High uncertainty.

Figure 2: Bayesian-LP: Impact of the number of samples on solution quality.



(a) Changing defense resources (20 voter groups). (b) Changing the # of voter groups (3 defense resources).

Figure 3: Scalability on synthetic data.



(a) Changing attack resources. (b) Changing defense resources.

Figure 4: Solution quality on real data

ods as well (DORABE). The results in Figure 3 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.

Finally, we evaluate our algorithms on the 2002 French president election dataset [Laslier and Van der Straeten, 2008], consisting of 2597 votes for 16 candidates by voters in 6 districts (voter groups). Figures 4(a) and 4(b) again compares the baselines to our algorithmic approach in terms of solution quality. As in the experiments with synthetic data, our approach demonstrates substantial improvement in defender's performance compared to baselines: in an extreme case, the attack success probability drops from 1 to nearly 0.

8 Conclusion

We study the problem of optimally protecting an election against group-deletion-control. We show that although plurality voting is vulnerable to control, it is NP-Hard to protect an election against it. We propose a double-oracle framework for computing an optimal protection strategy and develop compact mixed integer linear programs for both oracles, even though these are NP-Hard. We also propose heuristic oracles to further speed the double oracle framework up. Experimental results show that our algorithms outperform baseline alternatives, and scale to realistic problem instances.

Acknowledgments

This research was partially supported by the National Science Foundation (CNS-1238959, IIS-1526860), 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

[An *et al.*, 2013] Bo An, Matthew Brown, Yevgeniy Vorobeychik, and Milind Tambe. Security games with surveillance cost and optimal timing of attack execution. In *International Conference on Autonomous Agents and Multi-agent Systems*, pages 223–230, 2013.

[Bannet *et al.*, 2004] Jonathan Bannet, David W. Price, Algis Rudys, Justin Singer, and Dan S. Wallach. Hack-a-Vote: Security issues with electronic voting systems. *IEEE Security and Privacy*, 2(1):32–37, 2004.

[Bartholdi *et al.*, 1992] John J. Bartholdi, Craig A. Tovey, and Michael A. Trick. How hard is it to control an election? *Mathematical and Computer Modelling*, 16(8):27–40, 1992.

[Betzler and Uhlmann, 2009] Nadja Betzler and Johannes Uhlmann. Parameterized complexity of candidate control in elections and related digraph problems. *Theoretical Computer Science*, 410(52):5425–5442, 2009.

[Bhattacharjya, 2010] Satarupa Bhattacharjya. Low turnout and invalid votes mark first post war general polls. http://www.sundaytimes.lk/100411/News/nws_16.html, 2010.

[Chen *et al.*, 2014] Jiehua Chen, Piotr Faliszewski, Rolf Niedermeier, and Nimrod Talmon. Combinatorial voter control in elections. In *International Symposium on Mathematical Foundations of Computer Science*, pages 153–164, 2014.

[Chen *et al.*, 2015] Jiehua Chen, Piotr Faliszewski, Rolf Niedermeier, and Nimrod Talmon. Elections with few voters:

Candidate control can be easy. In *AAAI Conference on Artificial Intelligence*, pages 2045–2051, 2015.

[Conitzer and Sandholm, 2006] Vincent Conitzer and Tuomas Sandholm. Computing the optimal strategy to commit to. In *ACM Conference on Electronic Commerce*, pages 82–90, 2006.

[Erdélyi *et al.*, 2009] Gábor Erdélyi, Markus Nowak, and Jörg Rothe. Sincere-strategy preference-based approval voting fully resists constructive control and broadly resists destructive control. *Mathematical Logic Quarterly*, 55(4):425–443, 2009.

[Erdélyi *et al.*, 2015] Gábor Erdélyi, Edith Hemaspaandra, and Lane A. Hemaspaandra. More natural models of electoral control by partition. In *Algorithmic Decision Theory*, pages 396–413, 2015.

[Faliszewski *et al.*, 2011] Piotr Faliszewski, Edith Hemaspaandra, and Lane A Hemaspaandra. Multimode control attacks on elections. *Journal of Artificial Intelligence Research*, 40(1):305–351, 2011.

[Faliszewski *et al.*, 2013] Piotr Faliszewski, Edith Hemaspaandra, and Lane A. Hemaspaandra. Weighted electoral control. In *International Conference on Autonomous Agents and Multi-Agent Systems*, pages 367–374, 2013.

[Gan *et al.*, 2015] Jiarui Gan, Bo An, and Yevgeniy Vorobeychik. Security games with protection externality. In *AAAI Conference on Artificial Intelligence*, pages 914–920, 2015.

[Hemaspaandra *et al.*, 2007] Edith Hemaspaandra, Lane A. Hemaspaandra, and Jörg Rothe. Anyone but him: The complexity of precluding an alternative. *Artificial Intelligence*, 171(5):255–285, 2007.

[Hemaspaandra *et al.*, 2009] Edith Hemaspaandra, Lane A. Hemaspaandra, and Jörg Rothe. Hybrid elections broaden complexity-theoretic resistance to control. *Mathematical Logic Quarterly*, 55(4):397–424, 2009.

[Jain *et al.*, 2013] Manish Jain, Vincent Conitzer, and Milind Tambe. Security scheduling for real-world networks. In *International Conference on Autonomous Agents and Multi-Agent Systems*, pages 215–222, 2013.

[Korzhyk *et al.*, 2011a] Dmytro Korzhyk, Vincent Conitzer, and Ronald Parr. Security games with multiple attacker resources. In *International Joint Conference on Artificial Intelligence*, pages 273–279, 2011.

[Korzhyk *et al.*, 2011b] Dmytro Korzhyk, Zhengyu Yin, Christopher Kiekintveld, Vincent Conitzer, and Milind Tambe. Stackelberg vs. Nash in security games: An extended investigation of interchangeability, equivalence, and uniqueness. *Journal of Artificial Intelligence Research*, 41:297–327, 2011.

[Laslier and Van der Straeten, 2008] Jean-François Laslier and Karine Van der Straeten. A live experiment on approval voting. *Experimental Economics*, 11(1):97–105, 2008.

[Liu and Zhu, 2010] Hong Liu and Daming Zhu. Parameterized complexity of control problems in maximin election. *Information Processing Letters*, 110(10):383–388, 2010.

[Liu *et al.*, 2009] Hong Liu, Haodi Feng, Daming Zhu, and Junfeng Luan. Parameterized computational complexity of control problems in voting systems. *Theoretical Computer Science*, 410(27):2746–2753, 2009.

[McCormick, 1976] Garth P. McCormick. Computability of global solutions to factorable nonconvex programs: Part I - convex underestimating problems. *Mathematical Programming*, 10:147–175, 1976.

[Mcmahan *et al.*, 2003] H. Brendan Mcmahan, Geoffrey J Gordon, and Avrim Blum. Planning in the presence of cost functions controlled by an adversary. In *International Conference on Machine Learning*, pages 536–543, 2003.

[Menton, 2013] Curtis Menton. Normalized range voting broadly resists control. *Theory of Computing Systems*, 53(4):507–531, 2013.

[Parkes and Xia, 2012] David Parkes and Lirong Xia. A complexity-of-strategic-behavior comparison between Schulze's rule and ranked pairs. In *AAAI Conference on Artificial Intelligence*, pages 1429–1435, 2012.

[Paruchuri *et al.*, 2008] Praveen Paruchuri, Jonathan P. Pearce, Janusz Marecki, Milind Tambe, Fernando Ordóñez, and Sarit Kraus. Playing games with security: An efficient exact algorithm for Bayesian Stackelberg games. In *International Conference on Autonomous Agents and Multiagent Systems*, pages 895–902, 2008.

[RT, 2013] RT. Election day bombings sweep Pakistan: Over 30 killed, more than 200 injured. https://www.rt.com/news/pakistan-election-day-bombing-136/, 2013.

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

[Vorobeychik and Letchford, 2015] Yevgeniy Vorobeychik and Joshua Letchford. Securing interdependent assets. *Journal of the Autonomous Agents and Multiagent Systems*, 29(2):305–333, 2015.

[Wang *et al.*, 2016] Zhen Wang, Yue Yin, and Bo An. Computing optimal monitoring strategy for detecting terrorist plots. In *AAAI Conference on Artificial Intelligence*, 2016.

[Wolchok *et al.*, 2012] Scott Wolchok, Eric Wustrow, Dawn Isabel, and J. Alex Halderman. Attacking the Washington, DC internet voting system. In *International Conference on Financial Cryptography and Data Security*, pages 114–128, 2012.

[Xu *et al.*, 2008] Lin Xu, Frank Hutter, Holger H. Hoos, and Kevin Leyton-Brown. SATzilla: Portfolio-based algorithm selection for sat. *Journal of Artificial Intelligence Research*, 32(1):565–606, 2008.