A Parameterized Perspective on Protecting Elections

Palash Dey[1] , **Neeldhara Misra**[2] , **Swaprava Nath**[3] and **Garima Shakya**[3]

[1]Indian Institute of Technology, Kharagpur, India
[2]Indian Institute of Technology, Gandhinagar, India
[3]Indian Institute of Technology, Kanpur, India

palash.dey@cse.iitkgp.ac.in, neeldhara.m@iitgn.ac.in, {swaprava, garima}@cse.iitk.ac.in

Abstract

We study the parameterized complexity of the optimal defense and optimal attack problems in voting. In both the problems, the input is a set of voter groups (every voter group is a set of votes) and two integers k_a and k_d corresponding to respectively the number of voter groups the attacker can attack and the number of voter groups the defender can defend. A voter group gets removed from the election if it is attacked but not defended. In the optimal defense problem, we want to know if it is possible for the defender to commit to a strategy of defending at most k_d voter groups such that, no matter which k_a voter groups the attacker attacks, the outcome of the election does not change. In the optimal attack problem, we want to know if it is possible for the attacker to commit to a strategy of attacking k_a voter groups such that, no matter which k_d voter groups the defender defends, the outcome of the election is always different from the original (without any attack) one. We show that both the optimal defense problem and the optimal attack problem are computationally intractable for every scoring rule and the Condorcet voting rule even when we have only 3 candidates. We also show that the optimal defense problem for every scoring rule and the Condorcet voting rule is W[2]-hard for both the parameters k_a and k_d, while it admits a fixed parameter tractable algorithm parameterized by the combined parameter (k_a, k_d). The optimal attack problem for every scoring rule and the Condorcet voting rule turns out to be much harder – it is W[1]-hard even for the combined parameter (k_a, k_d). We propose two greedy algorithms for the OPTIMAL DEFENSE problem and empirically show that they perform effectively on reasonable voting profiles.

1 Introduction

The problem of election control asks if it is possible for an external agent, usually with a fixed set of resources, to influence the outcome of the election by altering its structure in some limited way. There are several specific manifestations of this problem: for instance, one may ask if it is possible to change the winner by deleting k voter groups, presumably by destroying ballot boxes or rigging electronically submitted votes. Indeed, several cases of violence at the ballot boxes have been placed on record [Bhattacharjya, 2010; RT, 2013], and in 2010, Halderman and his students exposed serious vulnerabilities in the electronic voting systems that are in widespread use in several states [Hal, 2010]. A substantial amount of the debates around the recently concluded presidential elections in the United States revolved around issues of potential fraud, with people voting multiple times, stuffing ballot boxes, etc. all of which are well recognized forms of election control. For example, [Wolchok *et al.*, 2012] studied security aspects on Internet voting systems.

The study of controlling elections is fundamental to computational social choice: it is widely studied from a theoretical perspective, and has deep practical impact. The pioneering work of [Bartholdi *et al.*, 1992] initiated the study of these problems from a computational perspective, hoping that computational hardness of these problems may suggest a substantial barrier to the phenomena of control: if it is, say NP-hard to control an election, then the manipulative agent may not be able to compute an optimal control strategy in a reasonable amount of time. This basic approach has been intensely studied in various other scenarios [Faliszewski *et al.*, 2011; Mattei *et al.*, 2014; Dey, 2018; Dey *et al.*, 2018; 2017; Dey and Misra, 2017; Dey *et al.*, 2015; Dey, 2019].

Exploring parameterized complexity of various control problems has also gained a lot of interest. For example, [Betzler and Uhlmann, 2009] studied parameterized complexity of candidate control in elections and showed interesting connection with digraph problems, [Liu and Zhu, 2010; 2013] studied parameterized complexity of control problem by deleting voters for many common voting rules, and so on [Dey *et al.*, 2016; 2019a]. Studying election control from a game theoretic approach using security games is also an active area of research. See, for example, the works of [An *et al.*, 2013; Letchford *et al.*, 2009].

The broad theme of using computational hardness as a barrier to control has two distinct limitations: one is, of course, that some voting rules simply remain computationally vulnerable to many forms of control, in the sense that optimal strategies can be found in polynomial time. The other is that even NP-hard control problems often admit reasonable heuristics, can be approximated well, or even admit efficient

Parameters	OPTIMAL DEFENSE		OPTIMAL ATTACK	
	Scoring rules	Condorcet	Scoring rules	Condorcet
k_d	W[2]-hard [Theorem 3]	W[2]-hard [Theorem 4]	W[2]-hard [Theorem 3]	W[2]-hard [Theorem 4]
k_a	W[2]-hard [Theorem 5]	W[2]-hard [Theorem 5]	W[1]-hard [Theorem 6]	W[1]-hard [Theorem 6]
(k_a, k_d)	$\mathcal{O}^*(k_a^{k_d})$ [Theorem 7] No poly kernel [Corollary 4]			
m	para-NP-hard [Corollary 3]		para-coNP-hard [Corollary 3]	

Table 1: Summary of parameterized complexity results. k_d : the maximum number of voter groups that the defender can defend. k_a : the maximum number of voter groups that the attacker can attack. m : the number of candidates.

exact algorithms in realistic scenarios. Therefore, relying on NP-hardness alone is arguably not a robust strategy against control. To address this issue, the work of [Yin *et al.*, 2016] explicitly defined the problem of *protecting an election from control*, where in addition to the manipulative agent, we also have a "defender", who can also deploy some resources to spoil a planned attack. In this setting, elections are defined with respect to *voter groups* rather than voters, which is a small difference from the traditional control setting. The voter groups model allows us to consider attacks on sets of voters, which is a more accurate model of realistic control scenarios.

In [Yin *et al.*, 2016], the defense problem is modeled as a Stackelberg game in which limited protection resources (say k_d) are deployed to protect a collection of voter groups and the adversary responds by attempting to subvert the election (by attacking, say, at most k_a groups). They consider the plurality voting rule, and show that the problem of choosing the minimal set of resources that guarantee that an election cannot be controlled is NP-hard. They further suggest a Mixed-Integer Program formulation that can usually be efficiently tackled by solvers. Our main contribution is to study this problem in a parameterized setting and provide a refined complexity landscape for it. We also introduce the complementary attack problem, and extend the study to voting rules beyond plurality. We now turn to a summary of our contributions.

1.1 Contribution

We refer the reader to Section 2 for the relevant formal definitions, while focusing here on a high-level overview of our results. Recall that the OPTIMAL DEFENSE problem asks for a set of at most k_d voter groups which, when protected, render any attack on at most k_a voter groups unsuccessful. In this paper, we study the parameterized complexity of OPTIMAL DEFENSE for all scoring rules and the Condorcet voting rule (these are natural choices because they are computationally vulnerable to control - - the underlying "attack problem" can be resolved in polynomial time). We show that the problem of finding an optimal defense is tractable when both the attacker and the defender have limited resources. Specifically, we show that the problem is fixed-parameter tractable with the combined parameter (k_a, k_d) by a natural bounded-depth search tree approach. We also show that the OPTIMAL DEFENSE problem is unlikely to admit a polynomial kernel under plausible complexity theoretic assumption. We observe that both these parameters are needed for fixed parameter tractability, as we show W[2]-hardness when OPTIMAL DEFENSE is parameterized by either k_a or k_d.

Another popular parameter considered for voting problems is m, the number of candidates — as this is usually small compared to the size of the election in traditional application scenarios. Unfortunately, we show that OPTIMAL DEFENSE is NP-hard even when the election has only 3 candidates, eliminating the possibility of fixed-parameter algorithms (and even XP algorithms). This strengthens a hardness result shown in [Yin *et al.*, 2016]. Our hardness results on a constant number of candidates rely on a succinct encoding of the information about the scores of the candidates from each voter group. We also observe that the problem is polynomially solvable when only two candidates are involved.

We introduce the complementary problem of attacking an election: here the attacker plays her strategy first, and the defender is free to defend any of the attacked groups within the budget. The attacker wins if she is successful in subverting the election no matter which defense is played out. This problem turns out to be harder: it is already W[1]-hard when parameterized by *both* k_a and k_d, which is in sharp contrast to the OPTIMAL DEFENSE problem. This problem is also hard in the setting of a constant number of candidates — specifically, it is coNP-hard for the plurality voting rule [Corollary 1] and the Condorcet voting rule [Corollary 2] even when we have only three candidates if every voter group is encoded as the number of plurality votes every candidate receives from that voter group. Our demonstration of the hardness of the attack problem is another step in the program of using computational intractability as a barrier to undesirable phenomenon, which, in this context, is the act of planning a systematic attack on voter groups with limited resources.

We finally propose two simple greedy algorithms for the OPTIMAL DEFENSE problem and empirically show that it may be able to solve many instances of practical interest.

2 Preliminaries

Let $\mathcal{C} = \{c_1, c_2, \ldots, c_m\}$ be a set of candidates and $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$ a set of voters. If not mentioned otherwise, we denote the set of candidates by $\mathcal{C}$, the set of voters by $\mathcal{V}$, the number of candidates by m, and the number of voters by n. Every voter v_i has a preference or vote $\succ_i$ which is a complete order over $\mathcal{C}$. We denote the set of all complete orders over $\mathcal{C}$ by $\mathcal{L}(\mathcal{C})$. We call a tuple of n preferences $(\succ_1, \succ_2, \cdots, \succ_n) \in \mathcal{L}(\mathcal{C})^n$ an n-voter preference profile. Often it is convenient to view a preference profile as a multi-set consisting of its votes. The view we are taking will be clear from the context. A voting rule (often called voting correspondence) is a function $r : \cup_{n\in\mathbb{N}}\mathcal{L}(\mathcal{C})^n \longrightarrow 2^{\mathcal{C}} \setminus \{\emptyset\}$ which selects, from a preference profile, a nonempty set of candidates as the winners. We refer the reader to [Brandt *et al.*, 2016] for a comprehensive introduction to computational social choice. In this paper we will be focusing on two voting rules – the scoring rules and the Condorcet voting rule which are defined as follows.

2.1 Scoring Rule

A collection of m-dimensional vectors $\overrightarrow{s_m} = (\alpha_1, \alpha_2, \ldots, \alpha_m) \in \mathbb{R}^m$ with $\alpha_1 \geqslant \alpha_2 \geqslant \ldots \geqslant \alpha_m$ and $\alpha_1 > \alpha_m$ for every $m \in \mathbb{N}$ naturally defines a voting rule — a candidate gets score α_i from a vote if it is placed at the i^{th} position, and the score of a candidate is the sum of the scores it receives from all the votes. The winners are the candidates with the highest score. Given a set of candidates $\mathcal{C}$, a score vector $\overrightarrow{\alpha}$ of length $|\mathcal{C}|$, a candidate $x \in \mathcal{C}$, and a profile $\mathcal{P}$, we denote the score of x in $\mathcal{P}$ by $s_{\mathcal{P}}^{\overrightarrow{\alpha}}(x)$. When the score vector $\overrightarrow{\alpha}$ is clear from the context, we omit $\overrightarrow{\alpha}$ from the superscript. A straight forward observation is that the scoring rules remain unchanged if we multiply every α_i by any constant $\lambda > 0$ and/or add any constant μ. Hence, we assume without loss of generality that for any score vector $\overrightarrow{s_m}$, there exists a j such that $\alpha_j - \alpha_{j+1} = 1$ and $\alpha_k = 0$ for all $k > j$. We call such a score vector a *normalized score vector*.

2.2 Weighted Majority Graph and Condorcet Voting Rule

Given an election $\mathcal{E} = (\mathcal{C}, (\succ_1, \succ_2, \ldots, \succ_n))$ and two candidates $x, y \in \mathcal{C}$, let us define $N_{\mathcal{E}}(x, y)$ to be the number of votes where the candidate x is preferred over y. We say that a candidate x defeats another candidate y in *pairwise election* if $N_{\mathcal{E}}(x, y) > N_{\mathcal{E}}(y, x)$. Using the election $\mathcal{E}$, we can construct a weighted directed graph $\mathcal{G}_{\mathcal{E}} = (\mathcal{U} = \mathcal{C}, E)$ as follows. The vertex set $\mathcal{U}$ of the graph $\mathcal{G}_{\mathcal{E}}$ is the set of candidates $\mathcal{C}$. For any two candidates $x, y \in \mathcal{C}$ with $x \neq y$, let us define the margin $\mathcal{D}_{\mathcal{E}}(x, y)$ of x from y to be $N_{\mathcal{E}}(x, y) - N_{\mathcal{E}}(y, x)$. We have an edge from x to y in $\mathcal{G}_{\mathcal{E}}$ if $\mathcal{D}_{\mathcal{E}}(x, y) > 0$. Moreover, in that case, the weight $w(x, y)$ of the edge from x to y is $\mathcal{D}_{\mathcal{E}}(x, y)$. A candidate c is called the *Condorcet winner* of an election $\mathcal{E}$ if there is an edge from c to every other vertices in the weighted majority graph $\mathcal{G}_{\mathcal{E}}$. The Condorcet voting rule outputs the Condorcet winner if it exists and outputs the set $\mathcal{C}$ of all candidates otherwise.

Let r be a voting rule. We study the r-OPTIMAL DEFENSE problem which was defined by [Yin *et al.*, 2016]. It is defined as follows. Intuitively, the r-OPTIMAL DEFENSE problem asks if there is a way to defend k_d voter groups such that, irrespective of which k_a voter groups the attacker attacks, the output of the election (that is the winning set of candidates) is always same as the original one. A voter group gets deleted if only if it is attacked but not defended.

Definition 1 (r-OPTIMAL DEFENSE). *Given n voter groups $\mathcal{G}_i, i \in [n]$, two integers k_a and k_d, does there exist an index set $\mathcal{I} \subseteq [n]$ with $|\mathcal{I}| \leqslant k_d$ such that, for every $\mathcal{I}' \subset [n]\backslash\mathcal{I}$ with $|\mathcal{I}'| \leqslant k_a$, we have $r((\mathcal{G}_i)_{i\in[n]\backslash\mathcal{I}'}) = r((\mathcal{G}_i)_{i\in[n]})$? The integers k_a and k_d are called respectively attacker's resource and defender's resource. We denote an arbitrary instance of the r-OPTIMAL DEFENSE problem by $(\mathcal{C}, \{\mathcal{G}_i : i \in [n]\}, k_a, k_d)$.*

We also study the r-OPTIMAL ATTACK problem which is defined as follows. Intuitively, in the r-OPTIMAL ATTACK problem the attacker is interested to know if it is possible to attack k_a voter groups such that, no matter which k_d voter groups the defender defends, the outcome of the election is never same as the original (that is the attack is successful).

Definition 2 (r-OPTIMAL ATTACK). *Given n voter groups $\mathcal{G}_i, i \in [n]$, two integers k_a and k_d, does there exist an index set $\mathcal{I} \subseteq [n]$ with $|\mathcal{I}| \leqslant k_a$ such that, for every $\mathcal{I}' \subseteq [n]$ with $|\mathcal{I}'| \leqslant k_d$, we have $r((\mathcal{G}_i)_{i\in[n]\backslash(\mathcal{I}\backslash\mathcal{I}')}) \neq r((\mathcal{G}_i)_{i\in[n]})$? We denote an arbitrary instance of the r-OPTIMAL ATTACK problem by $(\mathcal{C}, \{\mathcal{G}_i : i \in [n]\}, k_a, k_d)$.*

Encoding of the Input Instance: In both the r-OPTIMAL DEFENSE and r-OPTIMAL ATTACK problems, we assume that every input voter group $\mathcal{G}$ is encoded as follows. The encoding lists all the different votes $\succ$ that appear in the voter group $\mathcal{G}$ along with the number of times the vote $\succ$ appear in $\mathcal{G}$. Hence, if a voter group $\mathcal{G}$ contains only k different votes over m candidates and consists of n voters, then the encoding of $\mathcal{G}$ takes $\mathcal{O}(km \log m \log n)$ bits of memory.

2.3 Parameterized complexity

A parameterized problem Π is a subset of $\Gamma^* \times \mathbb{N}$, where Γ is a finite alphabet. A central notion is *fixed parameter tractability* (FPT) which means, for a given instance (x, k), solvability in time $f(k) \cdot p(|x|)$, where f is an arbitrary function of k and p is a polynomial in the input size $|x|$. There exists a hierarchy of complexity classes above FPT, and showing that a parameterized problem is hard for one of these classes is considered evidence that the problem is unlikely to be fixed-parameter tractable. The main classes in this hierarchy are: FPT $\subseteq$ W[1] $\subseteq$ W[2] $\subseteq \cdots \subseteq$ W[P] $\subseteq$ XP. We now define the notion of parameterized reduction [Cygan *et al.*, 2015].

Definition 3. *Let A, B be parameterized problems. We say that A is **fpt-reducible** to B if there exist functions $f, g : \mathbb{N} \to \mathbb{N}$, a constant $\alpha \in \mathbb{N}$ and an algorithm Φ which transforms an instance (x, k) of A into an instance $(x', g(k))$ of B in time $f(k)|x|^\alpha$ so that $(x, k) \in A$ if and only if $(x', g(k)) \in B$.*

To show W-hardness, it is enough to give a parameterized reduction from a known hard problem.

3 Classical Complexity Results

[Yin *et al.*, 2016] showed that the OPTIMAL DEFENSE problem is polynomial time solvable for the plurality voting rule when we have only 2 candidates. On the other hand, they also showed that the OPTIMAL DEFENSE problem is NP-complete when we have an *unbounded* number of candidates. We begin with improving their NP-completeness result by showing that the OPTIMAL DEFENSE problem becomes NP-complete even when we have only 3 candidates and the attacker can attack any number of voter groups. Towards that, we reduce the k-SUM problem to the OPTIMAL DEFENSE problem. The k-SUM problem is defined as follows.

Definition 4 (k-SUM). *Given a set of n positive integers $\mathcal{W} = \{w_i, i \in [n]\}$, and two positive integers $k \leqslant n$ and*

M, does there exist an index set $\mathcal{I} \subset [n]$ with $|\mathcal{I}| = k$ such that $\sum_{i\in\mathcal{I}} w_i = M$?

The k-SUM problem can be easily proved to be NP-complete by modifying the NP-completeness proof of the Subset Sum problem in [Cormen *et al.*, 2009]. We also need the following structural result for normalized scoring rules which has been used before [Baumeister *et al.*, 2011; Dey *et al.*, 2016].

Lemma 1. *Let $\mathcal{C} = \{c_1, \ldots, c_m\}$ be a set of candidates and $\overrightarrow{\alpha}$ a normalized score vector of length $|\mathcal{C}|$. Let $x, y \in \mathcal{C}, x \neq y$, be any two arbitrary candidates. Then there exists a profile $\mathcal{P}_x^y$ consisting of m votes such that we have the following.*
$s_{\mathcal{P}_x^y}(x) + 1 = s_{\mathcal{P}_x^y}(y) - 1 = s_{\mathcal{P}_x^y}(a)$ *for every* $a \in \mathcal{C} \setminus \{x, y\}$

For any two candidates $x, y \in \mathcal{C}, x \neq y$, we use $\mathcal{P}_x^y$ to denote the profile as defined in Lemma 1. We are now ready to present our NP-completeness result for the OPTIMAL DEFENSE problem for the scoring rules even in the presence of 3 candidates only. In the interest of space, we defer proof of some of our results to the full version of this paper [Dey *et al.*, 2019b].

Theorem 1. *The* OPTIMAL DEFENSE *problem is* NP*-complete for every scoring rule even if the number of candidates is* 3 *and the attacker can attack any number of the voter groups.*

Proof. For the interest of space, we only prove NP-hardness. Let $\overrightarrow{\alpha}$ be any normalized score vector of length 3. The OPTIMAL DEFENSE problem for the scoring rule based on $\overrightarrow{\alpha}$ belongs to NP. Let $(\mathcal{W} = \{w_1, \ldots, w_n\}, k, M)$ be an arbitrary instance of the k-SUM problem. We can assume, without loss of generality, that 8 divides M and w_i for every $i \in [n]$; if this is not the case, we replace M and w_i by respectively $8M$ and $8w_i$ for every $i \in [n]$ which clearly is an equivalent instance of the original instance. Let us also assume, without loss of generality, that $2k < n$ (if not then add enough copies of $M + 1$ to $\mathcal{W}$) and $M < \sum_{i=1}^n w_i$ (since otherwise, it is a trivial NO instance). We construct the following instance of the OPTIMAL DEFENSE problem for the scoring rule based on $\overrightarrow{\alpha}$. Let M' be an integer such that $M' > \sum_{i=1}^n w_i$ and 8 divides M'. We have 3 candidates, namely a, b, and c. We have the following voter groups.

– For every $i \in [n]$, we have a voter group $\mathcal{G}_i$ consisting of w_i copies of $\mathcal{P}_a^c$ (as defined in Lemma 1) and $M' - w_i$ copies of $\mathcal{P}_b^c$. Hence, we have the following.
$s_{\mathcal{G}_i}(c) = s_{\mathcal{G}_i}(a) + M' + w_i = s_{\mathcal{G}_i}(b) + 2M' - w_i$

– We have one voter group $\hat{\mathcal{G}}$ consisting of $(kM'+M)/2 - 3$ copies of $\mathcal{P}_c^a$, $(kM'-M)/2 - 1$ copies of $\mathcal{P}_c^b$, and $(kM'-M)/2 - 1$ copies of $\mathcal{P}_a^b$. We have the following.
$s_{\hat{\mathcal{G}}}(c) = s_{\hat{\mathcal{G}}}(a) - (kM' + M - 6) = s_{\hat{\mathcal{G}}}(b) - (2kM' - M - 6)$

Let $\mathcal{Q}$ be the resulting profile; that is $\mathcal{Q} = \cup_{i=1}^n \mathcal{G}_i \cup \hat{\mathcal{G}}$. We have $s_{\mathcal{Q}}(c) = s_{\mathcal{Q}}(a) + (n-k)M' + \sum_{i=1}^n w_i - M + 6 = s_{\mathcal{Q}}(b) + (n-2k)M' + M - \sum_{i=1}^n w_i + 6$. Since $n > 2k$ and $M' > \sum_{i=1}^n w_i$, we have $s_{\mathcal{Q}}(c) > s_{\mathcal{Q}}(a)$ and $s_{\mathcal{Q}}(c) > s_{\mathcal{Q}}(b)$. Thus the candidate c wins the election uniquely. We define k_d, the maximum number of voter groups that the defender can defend, to be k. We define k_a, the maximum number of voter groups that the attacker can attack, to be $n + 1$. This finishes the description of the OPTIMAL DEFENSE instance. We claim that the two instances are equivalent.

In the forward direction, let the k-SUM instance be a YES instance and $\mathcal{I} \subset [n]$ with $|\mathcal{I}| = k$ be an index set such that $\sum_{i\in\mathcal{I}} w_i = M$. Let us consider the defense strategy where the defender protects the voter groups $\mathcal{G}_i$ for every $i \in \mathcal{I}$. Since $\sum_{i\in\mathcal{I}} w_i = M$, we have $\sum_{i\in\mathcal{I}}(M' - w_i) = kM' - M$. Let $\mathcal{H}$ be the profile of voter groups corresponding to the index set $\mathcal{I}$; that is, $\mathcal{H} = \cup_{i\in\mathcal{I}} \mathcal{G}_i$. Let $\mathcal{H}'$ be the profile remaining after the attacker attacks some voter groups. Without loss of generality, we can assume that the attacker does not attack the voter group $\hat{\mathcal{G}}$ since otherwise the candidate c continues to win uniquely. We thus obviously have $\mathcal{H} \cup \hat{\mathcal{G}} \subseteq \mathcal{H}'$. We have $s_{\mathcal{H}\cup\hat{\mathcal{G}}}(c) = s_{\mathcal{H}\cup\hat{\mathcal{G}}}(a) + kM' + \sum_{i\in\mathcal{I}} w_i - (kM' + M - 6) = s_{\mathcal{H}\cup\hat{\mathcal{G}}}(a) + 6$ and $s_{\mathcal{H}\cup\hat{\mathcal{G}}}(c) = s_{\mathcal{H}\cup\hat{\mathcal{G}}}(b) + 2kM' - \sum_{i\in\mathcal{I}} w_i - (2kM' - M - 6) = s_{\mathcal{H}\cup\hat{\mathcal{G}}}(b) + 6$. Since the candidate c receives as much score as any other candidate in the voter group $\mathcal{G}_i$ for every $i \in [n]$, we have $s_{\mathcal{H}'\cup\hat{\mathcal{G}}}(c) \geqslant s_{\mathcal{H}'\cup\hat{\mathcal{G}}}(a) + 6$ and $s_{\mathcal{H}'\cup\hat{\mathcal{G}}}(c) \geqslant s_{\mathcal{H}'\cup\hat{\mathcal{G}}}(b) + 6$. Hence, the candidate c wins uniquely in the resulting profile $\mathcal{H}'$ after the attack and thus the defense is successful.

In the other direction, let the OPTIMAL DEFENSE instance be a YES instance. Without loss of generality, we can assume that the attacker does not attack the voter group $\hat{\mathcal{G}}$ and thus the defender does not defend the voter group $\hat{\mathcal{G}}$. We can also assume, without loss of generality, that the defender defends exactly k voter groups since the candidate c receives as much score as any other candidate in the voter group $\mathcal{G}_i$ for every $i \in [n]$. Let $\mathcal{I} \subset [n]$ with $|\mathcal{I}| = k$ such that defending all the voter groups $\mathcal{G}_i, i \in \mathcal{I}$ is a successful defense strategy. We claim that $\sum_{i\in\mathcal{I}} w_i \geqslant M$. Suppose not, then let us assume that $\sum_{i\in\mathcal{I}} w_i < M$. Since, w_i is divisible by 8 and positive for every $i \in [n]$ and m is divisible by 8, we have $\sum_{i\in\mathcal{I}} w_i \leqslant M - 8$. Let $\mathcal{H}$ be the profile of voter groups corresponding to the index set $\mathcal{I}$; that is, $\mathcal{H} = \cup_{i\in\mathcal{I}} \mathcal{G}_i$. We have $s_{\mathcal{H}\cup\hat{\mathcal{G}}}(c) = s_{\mathcal{H}\cup\hat{\mathcal{G}}}(a) + kM' + \sum_{i\in\mathcal{I}} w_i - (kM' + M - 6) \leqslant s_{\mathcal{H}\cup\hat{\mathcal{G}}}(a) + M - 8 - M + 6 = s_{\mathcal{H}\cup\hat{\mathcal{G}}}(a) - 2$. Hence attacking the voter groups $\mathcal{G}_i, i \in [n] \setminus \mathcal{I}$ makes the score of c strictly less than the score of a. This contradicts our assumption that defending all the voter groups $\mathcal{G}_i, i \in \mathcal{I}$ is a successful defense strategy. Hence we have $\sum_{i\in\mathcal{I}} w_i \geqslant M$. We now claim that $\sum_{i\in\mathcal{I}} w_i \leqslant M$. Suppose not, then let us assume that $\sum_{i\in\mathcal{I}} w_i > M$. Since, w_i is divisible by 8 and positive for every $i \in [n]$ and m is divisible by 8, we have $\sum_{i\in\mathcal{I}} w_i \geqslant M + 8$. Let $\mathcal{H}'$ be the profile of voter groups corresponding to the index set $\mathcal{I}$; that is, $\mathcal{H}' = \cup_{i\in\mathcal{I}} \mathcal{G}_i$. We have $s_{\mathcal{H}'\cup\hat{\mathcal{G}}}(c) = s_{\mathcal{H}'\cup\hat{\mathcal{G}}}(b) + 2kM' - \sum_{i\in\mathcal{I}} w_i - (2kM' - M - 6) \leqslant s_{\mathcal{H}'\cup\hat{\mathcal{G}}}(b) - (M + 8) + M + 6 = s_{\mathcal{H}'\cup\hat{\mathcal{G}}}(b) - 2$. Hence attacking the voter groups $\mathcal{G}_i, i \in [n] \setminus \mathcal{I}$ makes the score of c strictly less than the score of b. This contradicts our assumption that defending all the voter groups $\mathcal{G}_i, i \in \mathcal{I}$ is a successful defense strategy. Hence we have $\sum_{i\in\mathcal{I}} w_i \leqslant M$. Therefore we have $\sum_{i\in\mathcal{I}} w_i = M$ and thus the k-SUM instance is a YES instance. □

In the proof of Theorem 1, we observe that the reduced instance of the OPTIMAL DEFENSE problem viewed as an instance of the OPTIMAL ATTACK problem is a NO instance if and only if the k-SUM instance is a YES instance. Hence, the same reduction as in the proof of Theorem 1 gives us the following result for the OPTIMAL ATTACK problem.

Corollary 1. *The* OPTIMAL ATTACK *problem is* coNP*-hard for every scoring rule even if the number of candidates is* 3 *and the attacker can attack any number of voter groups.*

We now prove a similar hardness result as of Theorem 1 for the Condorcet voting rule.

Theorem 2. *The* OPTIMAL DEFENSE *problem is* NP*-complete for the Condorcet voting rule even if the number of candidates is* 3 *and the attacker can attack any number of voter groups.*

In the proof of Theorem 2, we observe that the reduced instance of OPTIMAL DEFENSE viewed as an instance of the OPTIMAL ATTACK problem is a NO instance if and only if the k-SUM instance is a YES instance. Hence, the same reduction as in the proof of Theorem 2 gives us the following result for the OPTIMAL ATTACK problem.

Corollary 2. *The* OPTIMAL ATTACK *problem is* coNP*-hard for the Condorcet voting rule even if the number of candidates is* 3 *and the attacker can attack any number of voter groups.*

4 W-Hardness Results

In this section, we present our hardness results for the OPTIMAL DEFENSE and the OPTIMAL ATTACK problems in the parameterized complexity framework. We consider the following parameters for both the problems – number of candidate (m), defender's resource (k_d), and attacker's resource (k_a). From Theorems 1 and 2 and Corollaries 1 and 2 we immediately have the following result for the OPTIMAL DEFENSE and OPTIMAL ATTACK problems parameterized by the number of candidates for both the scoring rules and the Condorcet voting rule.

Corollary 3. *The* OPTIMAL DEFENSE *problem is para-*NP*-hard parameterized by the number of candidates for both the scoring rules and the Condorcet voting rule. The* OPTIMAL ATTACK *problem is para-*coNP*-hard parameterized by the number of candidates for both the scoring rules and the Condorcet voting rule.*

The NP-completeness proof for the OPTIMAL DEFENSE problem for the plurality voting rule by [Yin *et al.*, 2016] is actually a parameter preserving reduction from the HITTING SET problem parameterized by the solution size.

Since the HITTING SET problem parameterized by the solution size k is known to be W[2]-complete [Downey and Fellows, 1999], the following result immediately follows from Theorem 2 of [Yin *et al.*, 2016].

Observation 1 ([Yin *et al.*, 2016])**.** *The* OPTIMAL DEFENSE *problem for the plurality voting rule is* W[2]*-hard parameterized by k_d.*

We now generalize Observation 1 to any scoring rule by exhibiting a polynomial parameter transform from the HITTING SET problem parameterized by the solution size.

Theorem 3. *The* OPTIMAL DEFENSE *and* OPTIMAL ATTACK *problems for every scoring rule is* W[2]*-hard parameterized by k_d.*

Next, we show the W[2]-hardness of the OPTIMAL DEFENSE and OPTIMAL ATTACK problems for the Condorcet voting rule parameterized by k_d. This is also a parameter-preserving reduction from the HITTING SET problem.

Theorem 4. *The* OPTIMAL DEFENSE *and* OPTIMAL ATTACK *problems for the Condorcet voting rule is* W[2]*-hard parameterized by k_d.*

We now show that the OPTIMAL DEFENSE problem for scoring rules is W[2]-hard parameterized by k_a also by exhibiting a parameter preserving reduction from a problem closely related to HITTING SET, which is SET COVER problem parameterized by the solution size. This is a W[2]-complete problem [Downey and Fellows, 1999]. We now state our W[2]-hardness proof for the OPTIMAL DEFENSE problem for scoring rules and the Condorcet voting rule, parameterized by k_a, by a reduction from SET COVER.

Theorem 5. *The* OPTIMAL DEFENSE *problem for every scoring rule and Condorcet rule is* W[2]*-hard parameterized by k_a.*

We now show that the OPTIMAL ATTACK problem for the scoring rules is W[1]-hard even parameterized by the combined parameter k_a and k_d. Towards that, we exhibit a polynomial parameter transform from the CLIQUE problem parameterized by the size of the clique we are looking for which is known to be W[1]-complete.

Theorem 6. *The* OPTIMAL ATTACK *problem for every scoring rule and Condorcet rule is* W[1]*-hard parameterized by (k_a, k_d).*

Once we have a parameterized algorithm for the OPTIMAL DEFENSE problem for the parameter (k_a, k_d), an immediate question is whether there exists a kernel for the OPTIMAL DEFENSE problem of size polynomial in (k_a, k_d). We know that the HITTING SET problem does not admit polynomial kernel parameterized by the universe size [Downey and Fellows, 1999]. It turns out that the reductions from the HITTING SET problem to the OPTIMAL DEFENSE and OPTIMAL ATTACK problems in Theorems 3, 4 and 6 are polynomial parameter transformations. Hence we immediately have the following corollary.

Corollary 4. *The* OPTIMAL DEFENSE *and* OPTIMAL ATTACK *problems for the scoring rules and the Condorcet rule do not admit a polynomial kernel parameterized by (k_a, k_d).*

5 The FPT Algorithm

We complement the negative results of Observation 1 and Theorem 5 by presenting an FPT algorithm for the OPTIMAL DEFENSE problem parameterized by (k_a, k_d). In the absence of a defender, that is when $k_d = 0$, [Yin *et al.*, 2016] showed that the OPTIMAL DEFENSE problem is polynomial time solvable for the plurality voting rule. Their polynomial time algorithm for the OPTIMAL DEFENSE problem can easily be extended to any scoring rule. Using this polynomial time algorithm, we design the following $\mathcal{O}^*(k_a^{k_d})$ time algorithm for the OPTIMAL DEFENSE problem for scoring rules.

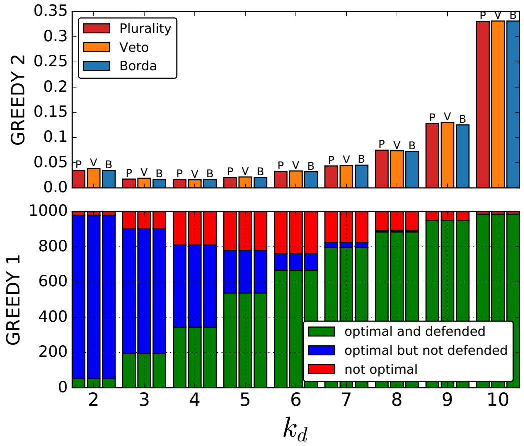

Figure 1: Performances of GREEDY 1 and GREEDY 2 for uniform voting profile generation model.

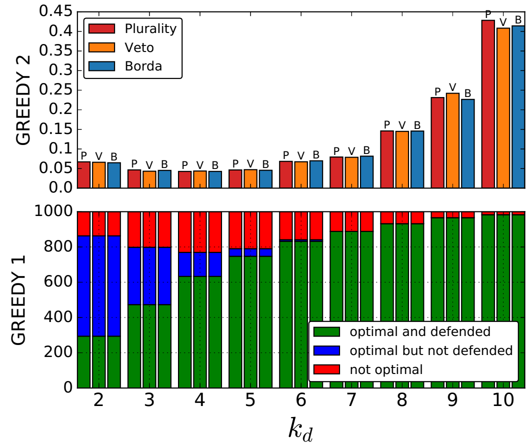

Figure 2: Performances of GREEDY 1 and GREEDY 2 for voting profile generation model with two major contesting candidates.

Theorem 7. *There is an algorithm for the* OPTIMAL DEFENSE *problem for every scoring rule and the Condorcet voting rule which runs in time* $\mathcal{O}^*(k_a^{k_d})$.

6 Experiments

Though the previous sections show that the optimal defending problem is computationally intractable, it is a worst-case result. In practice, elections have voting profiles that are generated from some (possibly known) distribution. In this section, we conduct an empirical study to understand how simple defending strategies perform for two such statistical voter generation models. The defending strategies we consider are variants of a simple greedy policy.

6.1 Defending Strategy

For a given voting profile and a voting rule, the defending strategy finds the winner. Suppose the winner is a. The strategy considers a with every other candidate, and for each such pair it creates a sorted list of classes based on the winning margin of votes for a in those classes, and picks the top k_d classes to form a sub-list. Now, among all these $(m-1)$ sorted sub-lists, the strategy picks the most frequent k_d classes to protect. We call this version of the strategy GREEDY 1. For certain profiles an optimal attacker (a) may change the outcome by attacking some of the unprotected classes or (b) is unable to change the outcome. If (a) occurs, then there is a possibility that for the value of k_d there does not exist any defense strategy which can guard the election from all possible strategies of the attacker. In that case, GREEDY 1 is optimal and is not optimal otherwise. It is always optimal for case (b). Note that, given a profile and k_d protected classes, it is easy to find if there exists an optimal attack strategy, while it is not so easy to identify whether there does not exist any defending strategy if the GREEDY 1 fails to defend. We find the latter with a brute-force search for this experiment. A small variant of GREEDY 1 is the following: when GREEDY 1 is unable to defend (which is possible to find out in poly-time), the strategy chooses to protect k_d classes uniformly at random. Call this strategy GREEDY 2.

6.2 Voting Profile Generation

Fix $m = 5$. We generate 1000 preference profiles over these alternatives for $n = 12000$, where each vote is picked uniformly at random from the set of all possible strict preference orders over m alternatives. The voters are partitioned into 12 classes containing equal number of voters. We consider three voting rules: plurality, veto, and Borda. The lower plot in Figure 1 shows the number of profiles which belongs to the three categories: (i) GREEDY 1 defends (is optimal), (ii) GREEDY 1 cannot defend but no defending strategy exists (is optimal), (iii) GREEDY 1 cannot defend but defending strategy exists (not optimal). The x-axis shows different values of k_d and we fix $k_a = 12 - k_d$.

The upper plot of Figure 1 shows the fraction of the profiles successfully defended by GREEDY 2 where GREEDY 1 is not optimal (i.e., cannot defend but defending strategy exists) when GREEDY 2 uniformly at random picks k_d classes 100 times. These fractions therefore serves as an empirical probability of successful defense of GREEDY 2 given GREEDY 1 is not optimal.

In an election where the primary contest happens between two major candidates, even though there are more candidates present, the generation model may be a little different. We also consider another generation model that generates 40% profiles having a fixed alternative a on top and the strict order of the $(m-1)$ alternatives is picked uniformly at random, a similar 40% profiles with some other alternative b on top, and the remaining 20% preferences are picked uniformly at random from the set of all possible strict preference orders. Similar experiments are run on this generation model and results are shown in Figure 2.

The results show that even though optimal defense is a hard problem, a simple strategy like greedy achieves more than 70% optimality. From the rest 30% non-optimal cases, the variant GREEDY 2 is capable of salvaging it into optimal with probability almost 5% for uniform generation model and above 5% for two-major contestant generation model for $k_d = k_a = 6$. This empirically hints at a possibility that defending real elections may not be too difficult.

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, AAMAS '13, Saint Paul, MN, USA, May 6-10, 2013*, pages 223–230, 2013.

[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.

[Baumeister *et al.*, 2011] Dorothea Baumeister, Magnus Roos, and Jörg Rothe. Computational complexity of two variants of the possible winner problem. In *The 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 853–860, 2011.

[Betzler and Uhlmann, 2009] Nadja Betzler and Johannes Uhlmann. Parameterized complexity of candidate control in elections and related digraph problems. *Theor. Comput. Sci.*, 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.

[Brandt *et al.*, 2016] Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel Procaccia. Handbook of computational social choice, 2016.

[Cormen *et al.*, 2009] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition, 2009.

[Cygan *et al.*, 2015] Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015.

[Dey and Misra, 2017] Palash Dey and Neeldhara Misra. On the exact amount of missing information that makes finding possible winners hard. In *Proc. 42nd International Symposium on Mathematical Foundations of Computer Science, MFCS*, pages 57:1–57:14, 2017.

[Dey *et al.*, 2015] Palash Dey, Neeldhara Misra, and Y. Narahari. Detecting possible manipulators in elections. In *Proc. International Conference on Autonomous Agents and Multiagent Systems, AAMAS*, pages 1441–1450, 2015.

[Dey *et al.*, 2016] Palash Dey, Neeldhara Misra, and Y. Narahari. Kernelization complexity of possible winner and coalitional manipulation problems in voting. *Theor. Comput. Sci.*, 616:111–125, 2016.

[Dey *et al.*, 2017] Palash Dey, Neeldhara Misra, and Y. Narahari. Frugal bribery in voting. *Theor. Comput. Sci.*, 676:15–32, 2017.

[Dey *et al.*, 2018] Palash Dey, Neeldhara Misra, and Y. Narahari. Complexity of manipulation with partial information in voting. *Theor. Comput. Sci.*, 726:78–99, 2018.

[Dey *et al.*, 2019a] Palash Dey, Neeldhara Misra, and Y. Narahari. Parameterized dichotomy of choosing committees based on approval votes in the presence of outliers. *Theor. Comput. Sci.*, 2019.

[Dey *et al.*, 2019b] Palash Dey, Neeldhara Misra, Swaprava Nath, and Garima Shakya. A parameterized perspective on protecting elections. *CoRR*, abs/1905.11838, 2019.

[Dey, 2018] Palash Dey. Manipulative elicitation - A new attack on elections with incomplete preferences. *Theor. Comput. Sci.*, 731:36–49, 2018.

[Dey, 2019] Palash Dey. Local distance restricted bribery in voting. In *Proc. 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS*, pages 1925–1927, 2019.

[Downey and Fellows, 1999] Rod G Downey and Michael Ralph Fellows. *Parameterized Complexity*, volume 3. springer Heidelberg, 1999.

[Faliszewski *et al.*, 2011] Piotr Faliszewski, Edith Hemaspaandra, and Lane A. Hemaspaandra. Multimode control attacks on elections. *J. Artif. Intell. Res. (JAIR)*, 40:305–351, 2011.

[Hal, 2010] Alex halderman strengthens democracy using software, Popular Science, http://www.popsci.com/brilliant-10-alex-halderman-strengthens-democracy-using-software, 2010.

[Letchford *et al.*, 2009] Joshua Letchford, Vincent Conitzer, and Kamesh Munagala. Learning and approximating the optimal strategy to commit to. In *Algorithmic Game Theory, Second International Symposium, SAGT 2009, Paphos, Cyprus, October 18-20, 2009. Proceedings*, pages 250–262, 2009.

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

[Liu and Zhu, 2013] Hong Liu and Daming Zhu. Parameterized complexity of control by voter selection in maximin, copeland, borda, bucklin, and approval election systems. *Theor. Comput. Sci.*, 498:115–123, 2013.

[Mattei *et al.*, 2014] Nicholas Mattei, Nina Narodytska, and Toby Walsh. How hard is it to control an election by breaking ties? In Torsten Schaub, Gerhard Friedrich, and Barry O'Sullivan, editors, *ECAI*, volume 263 of *Frontiers in Artificial Intelligence and Applications*, pages 1067–1068. IOS Press, 2014.

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

[Wolchok *et al.*, 2012] Scott Wolchok, Eric Wustrow, Dawn Isabel, and J. Alex Halderman. Attacking the washington, D.C. internet voting system. In *Financial Cryptography and Data Security - 16th International Conference, FC 2012, Kralendijk, Bonaire, Februray 27-March 2, 2012, Revised Selected Papers*, pages 114–128, 2012.

[Yin *et al.*, 2016] Yue Yin, Yevgeniy Vorobeychik, Bo An, and Noam Hazon. Optimally protecting elections. In *Proc. Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, pages 538–545, 2016.