# A Parameterized Perspective on Protecting Elections


**Palash Dey**
Indian Institute of Technology, Kharagpur, India
palash.dey@cse.iitkgp.ac.in

**Neeldhara Misra**
Indian Institute of Technology, Gandhinagar, India
neeldhara.m@iitgn.ac.in

**Swaprava Nath**
Indian Institute of Technology, Kanpur, India
swaprava@iitk.ac.in

**Garima Shakya**
Indian Institute of Technology, Kanpur, India
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 district consisting of 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 one (without any attack). 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 many voting profiles.



**Keywords and phrases** parameterized complexity, election control, optimal attack, optimal defense


## 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 [2, 8], and in 2010, Halderman and his students

<table>
<tr><th rowspan="2">Parameters</th><th colspan="2">OPTIMAL DEFENSE</th><th colspan="2">OPTIMAL ATTACK</th></tr>
<tr><th>Scoring rules</th><th>Condorcet</th><th>Scoring rules</th><th>Condorcet</th></tr>
<tr><td>$k_d$</td><td>W[2]-hard [Theorem 16]</td><td>W[2]-hard [Theorem 18]</td><td>W[2]-hard [Theorem 17]</td><td>W[2]-hard [Theorem 19]</td></tr>
<tr><td>$k_a$</td><td>W[2]-hard [Theorem 22]</td><td>W[2]-hard [Theorem 23]</td><td rowspan="2">W[1]-hard [Theorem 25]</td><td rowspan="2">W[1]-hard [Theorem 26]</td></tr>
<tr><td>$(k_a, k_d)$</td><td colspan="2">$\mathcal{O}^*(k_a^{k_d})$ [Theorem 28]<br>No poly kernel [Theorem 27]</td></tr>
<tr><td>$m$</td><td colspan="2">para-NP-hard [Theorem 14]</td><td colspan="2">para-coNP-hard [Theorem 14]</td></tr>
</table>

**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.

exposed serious vulnerabilities in the electronic voting systems that are in widespread use in several states [1]. 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. [67] 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. Bartholdi et al. [4] 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. For instance, Faliszewski et al. [30] studied the problem of control where different types of attacks are combined (multimode control), Mattei et al. [54] showed hardness of a variant of control which just exercises different tie-breaking rules, Bulteau et al. [11] and Kellerhals et al. [46] studied voter control in a combinatorial setting, etc [12, 15–18, 21, 22, 27, 27–29, 31–36, 38, 42, 43, 53, 55, 57–59, 61, 62].

Exploring parameterized complexity of various control problems has also gained a lot of interest. For example, Betzler and Uhlmann [7] studied parameterized complexity of candidate control in elections and showed interesting connection with digraph problems, Liu and Zhu [51, 52] studied parameterized complexity of control problem by deleting voters for many common voting rules, and so on [19, 23, 44, 47, 48, 50, 65]. 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. and Letchford et al. [3, 49].

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 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. [69] 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. [69], 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 denial-of-service(deletion) attack on (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.

**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 w.r.t. $(k_a, k_d)$ 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. [69]. 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 [Theorem 10] and the Condorcet voting rule [Theorem 13] 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.

### Organization:

The rest of the paper is organized as follows. In Section 2, we introduce necessary preliminaries; we present basic complexity results in Section 3; In Sections 4 and 5 we respectively present W-hardness results and fixed parameter tractable algorithms; we present our experimental findings in Section 6; we finally conclude in Section 7. A preliminary version of this work was published before [24] which did not contain most of the proofs.

## 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 [10] for a comprehensive introduction to computational social choice. In this paper we will be focusing on the scoring rules and the Condorcet voting rule which are defined as follows.

**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 $\alpha_i$ 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 $\alpha_i$ by any constant $\lambda > 0$ and/or add any constant $\mu$. 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*.

**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 and the voters are in $n$ disjoint partitions called voter groups. We study the $r$-Optimal Defense problem which was defined by Yin et al. [69]. 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 the same as the original one. A voter group gets deleted if only if it is attacked but not defended. More formally, the $r$-Optimal Defense problem is defined as follows.

▶ **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 $\mathcal{I}' \subset [n] \setminus \mathcal{I}$ with $|\mathcal{I}'| \leqslant k_a$, we have $r((\mathcal{G}_i)_{i \in [n] \setminus \mathcal{I}'}) = r((\mathcal{G}_i)_{i \in [n]})$? 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 $r((\mathcal{G}_i)_{i \in [n] \setminus (\mathcal{I} \setminus \mathcal{I}')}) \neq r((\mathcal{G}_i)_{i \in [n]})$? We denote an arbitrary instance of the $r$-Optimal Attack problem by $(\mathcal{C}, \{\mathcal{G}_i : i \in [n]\}, k_a, k_d)$.*

We observe that in the Optimal Defense problem, the defender first needs to commit to a defense strategy which the attacker can observe that they attack accordingly; the situation on the Optimal Attack problem is exactly reverse. Hence, a Yes-instance of Optimal Defense does not necessarily correspond to a No-instance of Optimal Attack.

**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.

**Parameterized complexity:** In parameterized complexity, each problem instance comes with a parameter $k$. Formally, a parameterized problem $\Pi$ is a subset of $\Gamma^* \times \mathbb{N}$, where $\Gamma$ is a finite alphabet. An instance of a parameterized problem is a tuple $(x, k)$, where $k$ is the parameter. 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 computable function of $k$ and $p$ is a polynomial in the input size $|x|$. The class FPT contains the fixed parameter tractable problems. Just as NP-hardness is used as evidence that a problem probably is not polynomial time solvable, 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 [14].

▶ **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* $\Phi$ *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 in the parameterized setting, it is enough to give a parameterized reduction from a known hard problem. We refer the reader to [14] for a detailed and formal introduction to parameterized complexity.

▶ **Definition 4. [Kernelization]** *[39, 40, 60] A kernelization algorithm for a parameterized problem* $\Pi \subseteq \Gamma^* \times \mathbb{N}$ *is an algorithm that, given* $(x, k) \in \Gamma^* \times \mathbb{N}$*, outputs, in time polynomial in* $|x| + k$*, a pair* $(x', k') \in \Gamma^* \times \mathbb{N}$ *such that (a)* $(x, k) \in \Pi$ *if and only if* $(x', k') \in \Pi$ *and (b)* $|x'|, k' \leqslant g(k)$*, where* $g$ *is some computable function. The output instance* $x'$ *is called the* kernel*, and the function* $g$ *is referred to as the* size of the kernel*. If* $g(k) = k^{O(1)}$*, then we say that* $\Pi$ *admits a polynomial kernel.*

For many parameterized problems, it is well established that the existence of a polynomial kernel would imply the collapse of the polynomial hierarchy to the third level (or more precisely, CoNP $\subseteq$ NP/Poly). Therefore, it is considered unlikely that these problems would admit polynomial-sized kernels. For showing kernel lower bounds, we simply establish reductions from these problems.

▶ **Definition 5. [Polynomial Parameter Transformation]** [9] *Let* $\Gamma_1$ *and* $\Gamma_2$ *be parameterized problems. We say that* $\Gamma_1$ *is polynomial time and parameter reducible to* $\Gamma_2$*, written* $\Gamma_1 \leqslant_{\text{PPT}} \Gamma_2$*, if there exists a polynomial time computable function* $f : \Sigma^* \times \mathbb{N} \to \Sigma^* \times \mathbb{N}$*, and a polynomial* $p : \mathbb{N} \to \mathbb{N}$*, and for all* $x \in \Sigma^*$ *and* $k \in \mathbb{N}$*, if* $f((x, k)) = (x', k')$*, then* $(x, k) \in \Gamma_1$ *if and only if* $(x', k') \in \Gamma_2$*, and* $k' \leqslant p(k)$*. We call* $f$ *a polynomial parameter transformation (or a* PPT*-reduction) from* $\Gamma_1$ *to* $\Gamma_2$*.*

This notion of a reduction is useful in showing kernel lower bounds because of the following theorem.

▶ **Theorem 6.** [9, Theorem 3] *Let* $P$ *and* $Q$ *be parameterized problems whose derived classical problems (i.e. the problems without parameters) are* $P^c, Q^c$*, respectively. Let* $P^c$ *be* NP*-complete, and* $Q^c \in$ NP*. Suppose there exists a PPT from* $P$ *to* $Q$*. Then, if* $Q$ *has a polynomial kernel, then* $P$ *also has a polynomial kernel.*

## 3 Classical Complexity Results

Yin et al. [69] 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 is 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 7** (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. [13]. We also need the following structural result for normalized scoring rules which has been used before [5, 20]. We include a proof of it for self-containment.

▶ **Lemma 8.** *Let $\mathcal{C} = \{c_1, \dots, c_m\}$ 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* $a \in \mathcal{C} \setminus \{x, y\}$

**Proof.** Since $\overrightarrow{\alpha} = (\alpha_1, \dots, \alpha_m)$ a normalized score vector, there exists a $1 \leqslant k \leqslant m-1$ such that $\alpha_k + 1 = \alpha_{k+1}$. Without loss of generality, let us assume that $x = c_1$ and $y = c_2$. Our profile $\mathcal{P}_x^y$ is obtained as follows. We take a vote $v_1$ where $x$ immediately follows $y$. We then keep "rotating right" and obtain $m-1$ other votes namely $v_2, \dots, v_m$. Clearly in the profile $v_1, \dots, v_m$ all the candidates receive the same score. Now we exchange the position of $x$ and $y$ in $v_k$ and obtain our desired profile $\mathcal{P}_x^y$ which is shown as follows.

$$
\begin{array}{ll}
v_1: & y \succ x \succ c_3 \succ \cdots \succ c_m \\
v_2: & c_m \succ y \succ x \succ \cdots \succ c_{m-1} \\
\cdots & \cdots \\
v_k': & \cdots \succ x \succ y \succ \cdots \\
\cdots & \cdots \\
v_m: & x \succ c_2 \succ c_3 \succ \cdots \succ y
\end{array}
$$

◀

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 8. 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.

▶ **Theorem 9.** *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.** The OPTIMAL DEFENSE problem for every scoring rule can be shown to belong to NP by using a defense strategy S (a subset of at most $k_d$ voter groups) as a certificate. The fact that the certificate can be validated in polynomial time involves checking if there exists a successful attack despite protecting all groups in S. This can be done in polynomial time as follows. Let $c$ be the winner of the election. We iterate over every non-winning candidate $x$ and delete $k_a$ unprotected voter groups which has maximum score of $c$ minus the score of $x$. We now turn to the reduction from k-SUM.

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 $(\mathcal{W} = \{w_1, \dots, w_n\}, k, M)$ 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 not, then 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 $\vec{\alpha}$. 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 8) 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 $\mathcal{G}_i, i \in \mathcal{J}$ is a successful defense strategy. Hence we have $\sum_{i\in\mathcal{J}} w_i \leqslant M$. Therefore we have $\sum_{i\in\mathcal{J}} w_i = M$ and thus the $k$-SUM instance is a YES instance. ◀

In the proof of Theorem 9, 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 9 gives us the following result for the OPTIMAL ATTACK problem.

▶ **Corollary 10.** *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 9 for the Condorcet voting rule. Towards that, we need the following lemma which has been used before [56,68]. We provide a proof for self-containment.

▶ **Lemma 11.** *For any function* $f : \mathcal{C} \times \mathcal{C} \longrightarrow \mathbb{Z}$*, such that*

1. $\forall a, b \in \mathcal{C}, f(a, b) = -f(b, a)$.
2. $\forall a, b, c, d \in \mathcal{C}, f(a, b) + f(c, d)$ *is even,*

*there exists a* $n$ *voters' profile such that for all* $a, b \in \mathcal{C}$*,* $a$ *defeats* $b$ *with a margin of* $f(a, b)$*. Moreover,*

$$n \text{ is even and } n = O\left(\sum_{\{a,b\}\in\mathcal{C}\times\mathcal{C}} |f(a, b)|\right)$$

**Proof.** The following 2 votes make candidate $a$ defeat $b$ by a margin of 2 and all other pair of candidates are tied.

$$a \succ b \succ c_3 \succ \cdots \succ c_m$$

$$c_m \succ \cdots \succ c_3 \succ a \succ b$$

Repeated application of the above block (for appropriate pairs of candidates) proves the result. ◀

▶ **Theorem 12.** *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.*

**Proof.** The OPTIMAL DEFENSE problem for the Condorcet voting rule clearly belongs to NP. To show NP-hardness, we reduce an arbitrary instance of the $k$-SUM problem to the OPTIMAL DEFENSE problem for the Condorcet voting rule. Let $(\{w_1, \ldots, w_n\}, k, M)$ be an arbitrary instance of the $k$-SUM problem. We construct the following instance of the OPTIMAL DEFENSE problem for the Condorcet voting rule. Let $M' = \max\{w_i : i \in [n]\}$. 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 $\mathcal{G}_i$ where $\mathcal{D}_{\mathcal{G}_i}(a, b) = 2w_i, \mathcal{D}_{\mathcal{G}_i}(a, c) = 2(M' - w_i)$, and $\mathcal{D}_{\mathcal{G}_i}(b, c) = 0$.
- We have one voter group $\hat{\mathcal{G}}$ where the candidates $b$ and $c$ receive respectively $\mathcal{D}_{\hat{\mathcal{G}}}(b, a) = 2M - 1, \mathcal{D}_{\hat{\mathcal{G}}}(c, a) = 2(kM' - M) - 1$, and $\mathcal{D}_{\hat{\mathcal{G}}}(b, c) = 1$.

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 $n+1$. We observe that the candidate $a$ is the Condorcet winner of the election. 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 $\mathcal{J} \subset [n]$ with $|\mathcal{J}| = k$ be an index set such that $\sum_{i \in \mathcal{J}} w_i = M$. Let us consider the defense strategy where the defender protects the voter groups $\mathcal{G}_i$ for every $i \in \mathcal{J}$. Since $\sum_{i \in \mathcal{J}} w_i = M$, we have $\sum_{i \in \mathcal{J}} (M' - w_i) = kM' - M$. Without loss of generality, we can assume that the attacker does not attack the voter group $\hat{\mathcal{G}}$. We observe that the candidate $a$ is the Condorcet winner of the election even when the attacker attacks all the voter groups $\mathcal{G}_j, j \in [n] \setminus \mathcal{J}$. Hence the OPTIMAL DEFENSE instance is a YES instance.

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; otherwise we extend the defense strategy to defend more groups so that exactly $k$ voter groups are defended. Let $\mathcal{J} \subset [n]$ with $|\mathcal{J}| = k$ such that defending all the voter groups $\mathcal{G}_i, i \in \mathcal{J}$ is a successful defense strategy. We claim that $\sum_{i \in \mathcal{J}} w_i \geqslant M$. Suppose not, then let us assume that $\sum_{i \in \mathcal{J}} w_i < M$. Then attacking the voter groups $\mathcal{G}_i, i \in [n] \setminus \mathcal{J}$ makes the candidate $b$ defeat the candidate $a$ in pairwise election. This contradicts our assumption that defending all the voter groups $\mathcal{G}_i, i \in \mathcal{J}$ is a successful defense strategy. Hence we have $\sum_{i \in \mathcal{J}} w_i \geqslant M$. We now claim that $\sum_{i \in \mathcal{J}} w_i \leqslant M$. Suppose not, then let us assume that $\sum_{i \in \mathcal{J}} w_i > M$. Then attacking the voter groups $\mathcal{G}_i, i \in [n] \setminus \mathcal{J}$ makes the candidate $c$ defeat the candidate $a$ in pairwise election. This contradicts our assumption that defending all the voter groups $\mathcal{G}_i, i \in \mathcal{J}$ is a successful defense strategy. Hence we have $\sum_{i \in \mathcal{J}} w_i \leqslant M$. Therefore we have $\sum_{i \in \mathcal{J}} w_i = M$ and thus the $k$-SUM instance is a YES instance. ◀

In the proof of Theorem 12, 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 12 gives us the following result for the OPTIMAL ATTACK problem.

▶ **Corollary 13.** *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 candidates ($m$), defender's resource ($k_d$), and attacker's resource ($k_a$). From Theorems 9, 10, 12 and 13 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 14.** *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. [69] is actually a parameter preserving reduction from the HITTING SET problem parameterized by the solution size. The HITTING SET problem is defined as follows.

▶ **Definition 15** (HITTING SET). *Given a universe $\mathcal{U}$, a set $\mathcal{S} = \{S_i : i \in [t]\}$ of subsets of $\mathcal{U}$, and a positive integer $k$ which is at most $|\mathcal{U}|$, does there exist a subset $\mathcal{W} \subseteq \mathcal{U}$ with $|\mathcal{W}| = k$ such that $\mathcal{W} \cap S_i \neq \emptyset$ for every $i \in [t]$. We denote an arbitrary instance of* HITTING SET *by* $(\mathcal{U}, \mathcal{S}, k)$.

Since the HITTING SET problem parameterized by the solution size $k$ is known to be W[2]-complete [25], the following result immediately follows from Theorem 2 of Yin et al. [69].

▷ **Observation 1 ( [69]).** 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 16.** *The* OPTIMAL DEFENSE *problem for every scoring rule is* W[2]*-hard parameterized by* $k_d$.

**Proof.** Let $(\mathcal{U}, \mathcal{S} = \{S_j : j \in [t]\}, k)$ be an arbitrary instance of HITTING SET. Let $\mathcal{U} = \{z_i : i \in [n]\}$. Without loss of generality, we assume that $S_j \neq \emptyset$ for every $j \in [t]$, since otherwise the instance is a NO instance. Let $\overrightarrow{\alpha}$ be a normalized score vector of length $t + 2$. We construct the following instance of the OPTIMAL DEFENSE problem for the scoring rule based on $\overrightarrow{\alpha}$. The set of candidates $\mathcal{C} = \{x_j : j \in [t]\} \cup \{y, d\}$. We have the following voter groups.

- For every $i \in [n]$, we have a voter group $\mathcal{G}_i$. For every $j \in [t]$ with $z_i \in S_j$ we have 2 copies of $\mathcal{P}_{x_j}^{d}$ in $\mathcal{G}_i$.
- We have one group $\hat{\mathcal{G}}$ where we have $2tn$ copies of $\mathcal{P}_{d}^{x_j}$ for every $j \in [n]$ and $2tn - 1$ copies of $\mathcal{P}_{d}^{y}$.

Let $\mathcal{Q}$ be the resulting profile; that is $\mathcal{Q} = \cup_{i=1}^{n} \mathcal{G}_i \cup \hat{\mathcal{G}}$. We define the defender's resource $k_d$ to be $k + 1$ and attacker's resource to be $n$. This finishes the description of the OPTIMAL DEFENSE instance. Since $S_j \neq \emptyset$ for every $j \in [t]$, we have $s_{\mathcal{Q}}(y) > s_{\mathcal{Q}}(x_j)$ for every $j \in [t]$. We also have $s_{\mathcal{Q}}(y) > s_{\mathcal{Q}}(d)$. Hence the candidate $y$ is the unique winner of the profile $\mathcal{Q}$. We now prove that the OPTIMAL DEFENSE instance $(\mathcal{C}, \mathcal{Q}, k_a, k_d)$ is equivalent to the HITTING SET instance $(\mathcal{U}, \mathcal{S}, k)$.

In the forward direction, let us suppose that the HITTING SET instance is a YES instance. Let $\mathcal{I} \subset [n]$ be such that $|\mathcal{I}| = k$ and $\{z_i : i \in \mathcal{I}\} \cap S_j \neq \emptyset$. We claim that the defender's strategy of defending the voter groups $\mathcal{G}_j$ for every $j \in [t] \setminus \mathcal{I}$ and $\hat{\mathcal{G}}$ results in a successful defense. 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. We thus obviously have $\mathcal{H} \cup \hat{\mathcal{G}} \subseteq \mathcal{H}'$. Since $\{z_i : i \in \mathcal{I}\}$ forms a hitting set, we have $s_{\mathcal{H}'}(y) > s_{\mathcal{H}'}(x_j)$ for every $j \in [t]$. Also since the voter group $\hat{\mathcal{G}}$ is defended, we have $s_{\mathcal{H}'}(y) > s_{\mathcal{H}'}(d)$. Hence the candidate $y$ continues to win uniquely even after the attack and hence the OPTIMAL DEFENSE instance is a YES instance.

In the other direction, let the OPTIMAL DEFENSE instance be a YES instance. Without loss of generality, we can assume that the defender defends the voter group $\hat{\mathcal{G}}$ since otherwise the attacker can attack the voter group $\hat{\mathcal{G}}$ which makes the score of the candidate $d$ more than the score of the candidate $y$ and thus defense would fail. We can also assume, without loss of generality, that the defender defends exactly $k$ voter groups. Let $\mathcal{I} \subset [n]$ with $|\mathcal{I}| = k$ such that defending all the voter groups $\mathcal{G}_i, i \in \mathcal{I}$ and $\hat{\mathcal{G}}$ is a successful defense strategy. Let us consider $\mathcal{Z} = \{z_i : i \in \mathcal{I}\} \subseteq \mathcal{U}$. We claim that $\mathcal{Z}$ must form a hitting set. Indeed, otherwise let us assume that there exists a $j \in [t]$ such that $\mathcal{Z} \cap S_j = \emptyset$. Consider the situation where the attacker attacks voter groups $\mathcal{G}_i$ for every $i \in [n] \setminus \mathcal{I}$. We observe that $s_{\cup_{i\in\mathcal{I}}\mathcal{G}_i \cup \hat{\mathcal{G}}}(x_j) > s_{\cup_{i\in\mathcal{I}}\mathcal{G}_i \cup \hat{\mathcal{G}}}(y)$. This contradicts our assumption that defending all the voter groups $\mathcal{G}_i, i \in \mathcal{I}$ and $\hat{\mathcal{G}}$ is a successful defense strategy. Hence $\mathcal{Z}$ forms a hitting set and thus the HITTING SET instance is a YES instance. ◀

In the proof of Theorem 16, 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 HITTING SET instance is a YES instance. Hence, the same reduction as in the proof of Theorem 16 gives us the following result for the OPTIMAL ATTACK problem.

▶ **Corollary 17.** *The* OPTIMAL ATTACK *problem for every scoring rule is* W[2]*-hard parameterized by* $k_d$.

We now show W[2]-hardness of the OPTIMAL DEFENSE problem for the Condorcet voting rule parameterized by $k_d$.

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

▶ **Theorem 18.** *The* OPTIMAL DEFENSE *problem for the Condorcet voting rule is* W[2]*-hard parameterized by* $k_d$.

**Proof.** Let $(\mathcal{U}, \mathcal{S} = \{S_j : j \in [t]\}, k)$ be an arbitrary instance of HITTING SET. Let $\mathcal{U} = \{z_i : i \in [n]\}$. Without loss of generality, we assume that $S_j \neq \emptyset$ for every $j \in [t]$ since otherwise the instance is a NO instance. We construct the following instance of the OPTIMAL DEFENSE problem for the Condorcet voting rule. The set of candidates $\mathcal{C} = \{x_j : j \in [t]\} \cup \{y\}$. For every $i \in [n]$, we have a voter group $\mathcal{G}_i$. For every $j \in [t]$ with $z_i \in S_j$ we have $\mathcal{D}_{\mathcal{G}_i}(y, x_j) = 2$. Let $\mathcal{Q}$ be the resulting profile; that is $\mathcal{Q} = \cup_{i=1}^{n} \mathcal{G}_i$. We define the defender's resource $k_d$ to be $k$ and attacker's resource to be $n$. This finishes the description of the OPTIMAL DEFENSE instance. Since $S_j \neq \emptyset$ for every $j \in [t]$, we have $\mathcal{D}_{\mathcal{Q}}(y, x_j) \geqslant 2$ for every $j \in [t]$. Hence the candidate $y$ is the Condorcet winner of the profile $\mathcal{Q}$. We now prove that the OPTIMAL DEFENSE instance $(\mathcal{C}, \mathcal{Q}, k_a, k_d)$ is equivalent to the HITTING SET instance $(\mathcal{U}, \mathcal{S}, k)$.

In the forward direction, let us suppose that the HITTING SET instance is a YES instance. Let $\mathcal{I} \subset [n]$ be such that $|\mathcal{I}| = k$ and $\{z_i : i \in \mathcal{I}\} \cap S_j \neq \emptyset$. We claim that the defender's strategy of defending the voter groups $\mathcal{G}_j$ for every $j \in [t] \setminus \mathcal{I}$ results in a successful defense. 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. We thus obviously have $\mathcal{H} \subseteq \mathcal{H}'$. Since $\{z_i : i \in \mathcal{I}\}$ forms a hitting set, we have $\mathcal{D}_{\mathcal{H}'}(y, x_j) \geqslant 2$ for every $j \in [t]$. Hence the candidate $y$ continues to win uniquely even after the attack and hence the OPTIMAL DEFENSE instance is a YES instance.

In the other direction, let the OPTIMAL DEFENSE instance be a YES instance. We can also assume, without loss of generality, that the defender defends exactly $k$ voter groups. 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. Let us consider $\mathcal{Z} = \{z_i : i \in \mathcal{I}\} \subseteq \mathcal{U}$. We claim that $\mathcal{Z}$ must form a hitting set. Indeed, otherwise let us assume that there exists a $j \in [t]$ such that $\mathcal{Z} \cap S_j = \emptyset$. Consider the situation where the attacker attacks voter groups $\mathcal{G}_i$ for every $i \in [n] \setminus \mathcal{I}$. We observe that $\mathcal{D}_{\cup_{i \in \mathcal{I}} \mathcal{G}_i}(y, x_j) = 0$ and hence the candidate $y$ is not the Condorcet winner. This contradicts our assumption that defending all the voter groups $\mathcal{G}_i, i \in \mathcal{I}$ is a successful defense strategy. Hence $\mathcal{Z}$ forms a hitting set and thus the HITTING SET instance is a YES instance. ◀

In the proof of Theorem 18, 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 HITTING SET instance is a YES instance. Hence, the same reduction as in the proof of Theorem 18 gives us the following result for the OPTIMAL ATTACK problem.

▶ **Corollary 19.** *The* OPTIMAL ATTACK *problem 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. The SET COVER problem is defined as follows.

▶ **Definition 20** (SET COVER). *Given a universe $\mathcal{U}$, a set $\mathcal{S} = \{S_i : i \in [t]\}$ of subsets of $\mathcal{U}$, and a non-negative integer $k$ which is at most $t$, does there exist an index set $\mathcal{I} \subset [t]$ with $|\mathcal{I}| = k$ such that $\bigcup_{i \in \mathcal{I}} S_i = \mathcal{U}$. We denote an arbitrary instance of* SET COVER *by $(\mathcal{U}, \mathcal{S}, k)$.*

This is a W[2]-complete problem [25]. We now present our W[2]-hardness proof for the OPTIMAL DEFENSE problem for scoring rules parameterized by $k_a$, by a reduction from SET COVER.

▶ **Theorem 21.** *The* OPTIMAL DEFENSE *problem for every scoring rule and Condorcet rule is* W[2]*-hard parameterized by* $k_a$.

▶ **Theorem 22.** *The* OPTIMAL DEFENSE *problem for every scoring rule is* W[2]*-hard parameterized by* $k_a$.

**Proof.** Let $(\mathcal{U}, \mathcal{S} = \{S_j : j \in [t]\}, k)$ be an arbitrary instance of SET COVER. Let $\mathcal{U} = \{z_i : i \in [n]\}$. We assume that $k > 3$ since otherwise the SET COVER instance is polynomial time solvable. For $i \in [n]$, let $f_i$ be the number of $j \in [t]$ such that $z_i \in S_j$; that is, $f_i = |\{j \in [t] : z_i \in S_j\}|$. We assume, without loss of generality, that for every $i \in [n]$, $t - f_i - k > 3k$ by adding at most $9t$ empty sets in $\mathcal{S}$. We construct the following instance of the OPTIMAL DEFENSE problem for the scoring rule induced by the score vector $\overrightarrow{\alpha}$. The set of candidates $\mathcal{C} = \{x_i : i \in [n]\} \cup \{y, d\}$. Let $\overrightarrow{\alpha}$ be any normalized score vector of length $n + 2$. We have the following voter groups.

- For every $j \in [t]$, we have a voter group $\mathcal{G}_j$. For every $i \in [n]$ and $j \in [t]$ with $z_i \notin S_j$, we have 2 copies of $\mathcal{P}_{x_i}^{d}$.
- We have another voter group $\mathcal{H}$ where, for every $i \in [n]$, we have $2tn + (2(t - f_i - k) + 1)$ copies of $\mathcal{P}_{d}^{x_i}$ and $2tn$ copies of $\mathcal{P}_{d}^{y}$.

We define attacker resource $k_a$ to be $k$ and the defender's resource $k_d$ to be $t-k$. This finishes the description of the OPTIMAL DEFENSE instance. We first observe that the score of the candidate $d$ is strictly less than the score of every other candidate. We now observe that the candidate $y$ is the unique winner of the election since the score of the candidate $y$ is $2k-1$ more than the score of the candidate $x_i$ for every $i \in [n]$. We now prove that the OPTIMAL DEFENSE instance $(\mathcal{C}, \cup_{j\in[t]}\mathcal{G}_j \cup \mathcal{H}, k_a, k_d)$ is equivalent to the SET COVER instance $(\mathcal{U}, \mathcal{S}, k)$.

In the forward direction, let us suppose that the SET COVER instance is a YES instance. Let $\mathcal{J} \subset [t]$ be such that $|\mathcal{J}| = k$ and $\bigcup_{j\in\mathcal{J}} S_j = \mathcal{U}$. We claim that the defender's strategy of defending the voter groups $\mathcal{G}_j$ for every $j \in [t] \setminus \mathcal{J}$ results in a successful defense. To see this, we first observe that, if the attacker attacks the voter group $\mathcal{H}$, then the candidate $y$ continues to uniquely win the election irrespective of what other voter groups the attacker attacks. Indeed, since $t - f_i - k > 3k$ for every $i \in [n]$, the score of the candidate $x_i$ is strictly less than the score of the candidate $y$ irrespective of what other voter groups the attacker attacks. Since, for every $i \in [n]$ and $j \in [t]$, the score of the candidate $x_i$ is not more than the score of the candidate $y$ in the voter group $\mathcal{G}_j$, we may assume that the attacker attacks the voter group $\mathcal{G}_j$ for every $j \in \mathcal{J}$ (since they are the only voter groups unprotected except $\mathcal{H}$). Now, since $S_j, j \in \mathcal{J}$ forms a set cover of $\mathcal{U}$, after deleting the voter groups $\mathcal{G}_j, j \in \mathcal{J}$, the score of the candidate $x_i$ increases by at most $2(k-1)$ from the original election for every $i \in [n]$. Hence, after deleting the voter groups $\mathcal{G}_j, j \in \mathcal{J}$, the score of the candidate $x_i$ is still strictly less than the score of the candidate $y$. Hence the candidate $y$ continues to win and thus the defense is successful. Hence the OPTIMAL DEFENSE instance is a YES instance.

In the other direction, let us suppose that the OPTIMAL DEFENSE instance is a YES instance. We assume, without loss of generality, that the defender protects exactly $t-k$ voter groups. We argued in the forward direction that we can assume, without loss of generality, that the attacker never attacks the voter group $\mathcal{H}$. Hence, we can also assume, without loss of generality, that the defender also does not defend the voter group $\mathcal{H}$. Let $\mathcal{J} \subset [t]$ be such that $|\mathcal{J}| = k$ and the defender defends the voter group $\mathcal{G}_j$ for every $j \in [t] \setminus \mathcal{J}$. We claim that the sets $S_j, j \in \mathcal{J}$ forms a set cover of $\mathcal{U}$. Suppose not, then let $z_i$ be an element in $\mathcal{U}$ which is not covered by $S_j, j \in \mathcal{J}$. We observe that attacking the voter groups $\mathcal{G}_j$ for every $j \in \mathcal{J}$ increases the score of the candidate $x_i$ by $2k$ which makes the candidate $y$ lose in the resulting election (after deleting the voter groups $\mathcal{G}_j$ for every $j \in \mathcal{J}$) since the score of $x_i$ is strictly more than the score of $y$. This contradicts our assumption that defending the voter group $\mathcal{G}_j$ for every $j \in [t] \setminus \mathcal{J}$ is a successful defense strategy. Hence $S_j, j \in \mathcal{J}$ forms a set cover of $\mathcal{U}$ and thus the SET COVER instance is a YES instance. ◀

We now present our W[2]-hardness proof for the OPTIMAL DEFENSE problem for the Condorcet voting rule parameterized by $k_a$.

► **Theorem 23.** *The* OPTIMAL DEFENSE *problem for the Condorcet voting rule is* W[2]*-hard parameterized by* $k_a$.

**Proof.** Let $(\mathcal{U}, \mathcal{S} = \{S_j : j \in [t]\}, k)$ be an arbitrary instance of SET COVER. Let $\mathcal{U} = \{z_i : i \in [n]\}$. We assume that $k > 3$ since otherwise the SET COVER instance is polynomial time solvable. For $i \in [n]$, let $f_i$ be the number of $j \in [t]$ such that $z_i \in S_j$; that is, $f_i = |\{j \in [t] : z_i \in S_j\}|$. We assume, without loss of generality, that for every $i \in [n]$, $t - f_i - k > 3k$ by adding at most $9t$ empty sets in $\mathcal{S}$. We construct the following instance of the Optimal Defense problem for the Condorcet voting rule. The set of candidates $\mathcal{C} = \{x_i : i \in [n]\} \cup \{y\}$. We have the following voter groups.

- For every $j \in [t]$, we have a voter group $\mathcal{G}_j$. For every $i \in [n]$ and $j \in [t]$, we have $\mathcal{D}_{\mathcal{G}_j}(y, x_i) = 2$ if $z_i \notin S_j$ and $\mathcal{D}_{\mathcal{G}_j}(y, x_i) = 0$ otherwise. We also have $\mathcal{D}_{\mathcal{G}_j}(x_i, x_\ell) = 0$ for every $j \in [t], i, \ell \in [n]$ with $i \neq \ell$.
- We have another voter group $\mathcal{H}$ where, for every $i \in [n]$, we have $\mathcal{D}_{\mathcal{H}}(x_i, y) = 2(t - f_i - k)$. We also have $\mathcal{D}_{\mathcal{H}}(x_i, x_\ell) = 0$ for every $i, \ell \in [n]$ with $i \neq \ell$.

We define attacker resource $k_a$ to be $k$ and the defender's resource $k_d$ to be $t - k$. This finishes the description of the Optimal Defense instance. We first observe that the candidate $y$ is a Condorcet winner of the resulting election. We now prove that the Optimal Defense instance $(\mathcal{C}, \cup_{j \in [t]} \mathcal{G}_j \cup \mathcal{H}, k_a, k_d)$ is equivalent to the Set Cover instance $(\mathcal{U}, \mathcal{S}, k)$.

In the forward direction, let us suppose that the given Set Cover is a Yes instance. Let $\mathcal{J} \subset [t]$ be such that $|\mathcal{J}| = k$ and $\bigcup_{j \in \mathcal{J}} S_j = \mathcal{U}$. We claim that the defender's strategy of defending the voter groups $\mathcal{G}_j$ for every $j \in [t] \setminus \mathcal{J}$ results in a successful defense. To see this, we first observe that, we can assume without loss of generality that the attacker does not attack the voter group $\mathcal{H}$ since the candidate $y$ loses every pairwise election in $\mathcal{H}$. Since, for every $i \in [n]$ and $j \in [t]$, the candidate $y$ does not lose any pairwise election in the voter group $\mathcal{G}_j$, we may assume that the attacker attacks the voter group $\mathcal{G}_j$ for every $j \in \mathcal{J}$ (since they are the only voter groups unprotected except $\mathcal{H}$). Now, since $S_j, j \in \mathcal{J}$ forms a set cover of $\mathcal{U}$, after deleting the voter groups $\mathcal{G}_j, j \in \mathcal{J}$, we have $\mathcal{D}_{\cup_{j \in [t] \setminus \mathcal{J}} \mathcal{G}_i \cup \mathcal{H}}(y, x_i) \geqslant 2(t - f_i - k + 1) - 2(t - f_i - k) = 2$ for every $i \in [n]$. Hence, after deleting the voter groups $\mathcal{G}_j, j \in \mathcal{J}$, the candidate $y$ continues to be the Condorcet winner of the remaining profile. Hence the Optimal Defense instance is a Yes instance.

In the other direction, let us suppose that the contructed Optimal Defense instance is a Yes instance. We assume, without loss of generality, that the defender protects exactly $t - k$ voter groups. We argued in the forward direction that we can assume, without loss of generality, that the attacker never attacks the voter group $\mathcal{H}$. Hence, we can also assume, without loss of generality, that the defender also does not defend the voter group $\mathcal{H}$. Let $\mathcal{J} \subset [t]$ be such that $|\mathcal{J}| = k$ and the defender defends the voter group $\mathcal{G}_j$ for every $j \in [t] \setminus \mathcal{J}$. We claim that the sets $S_j, j \in \mathcal{J}$ forms a set cover of $\mathcal{U}$. Suppose not, then let $z_i$ be an element in $\mathcal{U}$ which is not covered by $S_j, j \in \mathcal{J}$. We observe that $\mathcal{D}_{\cup_{j \in [t] \setminus \mathcal{J}} \mathcal{G}_i \cup \mathcal{H}}(y, x_i) = 2(t - f_i - k) - 2(t - f_i - k) = 0$ and thus attacking the voter groups $\mathcal{G}_j$ for every $j \in \mathcal{J}$ makes the candidate $y$ not the Condorcet winner. This contradicts our assumption that defending the voter group $\mathcal{G}_j$ for every $j \in [t] \setminus \mathcal{J}$ is a successful defense strategy. Hence $S_j, j \in \mathcal{J}$ forms a set cover of $\mathcal{U}$ and thus the Set Cover instance is a Yes instance. ◀

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. The Clique problem is defined as follows.

▶ **Definition 24** (Clique). *Given a graph $\mathcal{G}$ and an integer $k$, does there exist a clique in $\mathcal{G}$ of size $k$? We denote an arbitrary instance of* Clique *by $(\mathcal{G}, k)$.*

▶ **Theorem 25.** *The* Optimal Attack *problem for every scoring rule is* W[1]*-hard parameterized by $(k_a, k_d)$.*

**Proof.** Let $(\mathcal{G} = (\mathcal{V}, \mathcal{E}), k)$ be an arbitrary instance of the CLIQUE problem. Let $\mathcal{V} = \{v_i : i \in [n]\}$ and $\mathcal{E} = \{e_j : j \in [m]\}$. Let $\overrightarrow{\alpha}$ be any arbitrary normalized score vector of length $m+2$. We construct the following instance of the OPTIMAL ATTACK problem for the scoring rule induced by the score vector $\overrightarrow{\alpha}$. The set of candidates $\mathcal{C} = \{x_j : j \in [m]\} \cup \{y, d\}$. We have the following voter groups.

- For every $i \in [n]$, we have a voter group $\mathcal{G}_i$. For every $i \in [n]$, we have $10m$ copies of $\mathcal{P}_d^x$ for every $x \in \mathcal{C} \setminus \{d\}$ in $\mathcal{G}_i$. We also have two copies of $\mathcal{P}_{x_j}^d$ in the voter group $\mathcal{G}_i$ if the edge $e_j$ is incident on the vertex $v_i$, for every $i \in [n]$ and $j \in [m]$.
- We have another voter group $\mathcal{H}$. We have one copy of $\mathcal{P}_d^{x_j}$ for every $j \in [m]$ in $\mathcal{H}$.

We define attacker resource $k_a$ to be $k$ and the defender's resource $k_d$ to be $k-2$. This finishes the description of the OPTIMAL ATTACK instance. Let $\mathcal{Q}$ be the resulting profile; that it $\mathcal{Q} = \cup_{i \in [n]} \mathcal{G}_i \cup \mathcal{H}$. We first observe that the candidate $y$ is the winner of the resulting election since $s_{\mathcal{Q}}(y) = s_{\mathcal{Q}}(x_j) + 3$ and $s_{\mathcal{Q}}(y) > s_{\mathcal{Q}}(d)$. We now prove that the OPTIMAL ATTACK instance $(\mathcal{C}, \mathcal{Q}, k_a, k_d)$ is equivalent to the CLIQUE instance $(\mathcal{G}, k)$.

In the forward direction, let us assume that $\mathcal{U} = \{v_i : i \in \mathcal{I}\} \subset \mathcal{V}$ with $|\mathcal{I}| = k$ forms a clique in $\mathcal{G}$. We claim that attacking all the voter groups $\mathcal{G}_i, i \in \mathcal{I}$ forms a successful attack. Indeed, suppose the defender defends all the voter groups $\mathcal{G}_i, i \in \mathcal{I}$ except $\mathcal{G}_\ell$ and $\mathcal{G}_{\ell'}$ for some $\ell, \ell' \in \mathcal{I}$ with $\ell \neq \ell'$; they must exist since $k_d = k_a - 2$. Let $e_{j^\star}$ be the edge between the vertices $v_\ell$ and $v_{\ell'}$ in $\mathcal{G}$. Let the profile after the attack be $\hat{\mathcal{G}}$; that is, $\hat{\mathcal{G}} = \cup_{i \in [n] \setminus \mathcal{I}} \mathcal{G}_i \cup \mathcal{G}_\ell \cup \mathcal{G}_{\ell'} \cup \mathcal{H}$. Then we have $s_{\hat{\mathcal{G}}}(y) = s_{\hat{\mathcal{G}}}(x_{j^\star}) - 1$ and thus the candidate $y$ does not win after the attack. Hence the OPTIMAL ATTACK instance is YES instance.

In the other direction, let the OPTIMAL ATTACK instance be a YES instance. We first observe that the candidate $d$ performs worse than everyone else in every voter group and thus $d$ can never win. Now we can assume, without loss of generality, that the attacker does not attack the voter group $\mathcal{H}$ since the candidate $y$ is not receiving more score than any other candidate except $d$ in $\mathcal{H}$. Let attacking all the voter groups $\mathcal{G}_i, i \in \mathcal{I}$ with $|\mathcal{I}| \leqslant k$ is a successful attack. We observe that if $|\mathcal{I}| < k$, then defending any $k-2$ of the groups that are attacked foils the attack – since the candidate $y$ continues to win even after deleting any one group. Hence we have $|\mathcal{I}| = k$. Let us consider the subset of vertices $\mathcal{U} = \{v_i : i \in \mathcal{I}\}$. We claim that $\mathcal{U}$ forms a clique in $\mathcal{G}$. Indeed, if not, then let us assume that there exists two indices $\ell, \ell' \in \mathcal{I}$ such that there is no edge between the vertices $v_\ell$ and $v_{\ell'}$ in $\mathcal{G}$. Let us consider the defender strategy of defending all the voter groups $\mathcal{G}_i, i \in \mathcal{I} \setminus \{\ell, \ell'\}$. We observe that the candidate $y$ continues to uniquely receive the highest score among all the candidates and thus $y$ wins uniquely in the resulting election. This contradicts our assumption that attacking all the voter groups $\mathcal{G}_i, i \in \mathcal{I}$ with $|\mathcal{I}| \leqslant k$ is a successful attack. Hence $\mathcal{U}$ forms a clique in $\mathcal{G}$ and thus the CLIQUE instance is a YES instance. ◀

We now show similar result as of Theorem 25 for the Condorcet voting rule.

▶ **Theorem 26.** *The* OPTIMAL ATTACK *problem for the Condorcet voting rule is* W[1]*-hard parameterized by* $(k_a, k_d)$.

**Proof.** Let $(\mathcal{G} = (\mathcal{V}, \mathcal{E}), k)$ be an arbitrary instance of the CLIQUE problem. Let $\mathcal{V} = \{v_i : i \in [n]\}$ and $\mathcal{E} = \{e_j : j \in [m]\}$. We construct the following instance of the OPTIMAL ATTACK problem for the Condorcet voting rule. The set of candidates $\mathcal{C} = \{x_j : j \in [m]\} \cup \{y\}$. We have the following voter groups.

– For every $i \in [n]$, we have a voter group $\mathcal{G}_i$. We have $\mathcal{D}_{\mathcal{G}_i}(y, x_j) = 2$ if the edge $e_j$ is incident on the vertex $v_i$ and $\mathcal{D}_{\mathcal{G}_i}(y, x_j) = 0$ if the edge $e_j$ is not incident on the vertex $v_i$, for every $i \in [n]$ and $j \in [m]$. We also have $\mathcal{D}_{\mathcal{G}_i}(x_\ell, x_j) = 0$ for every $i \in [n]$, $j, \ell \in [m]$, and $j \neq \ell$.
– We have another voter group $\mathcal{H}$ where we have $\mathcal{D}_{\mathcal{H}}(x_j, y) = 2$ for every $j \in [m]$ and $\mathcal{D}_{\mathcal{H}}(x_\ell, x_j) = 0$ for every $j, \ell \in [m]$ and $j \neq \ell$.

We define attacker resource $k_a$ to be $k$ and the defender's resource $k_d$ to be $k - 2$. This finishes the description of the Optimal Attack instance. Let $\mathcal{Q}$ be the resulting profile; that is $\mathcal{Q} = \cup_{i \in [n]} \mathcal{G}_i \cup \mathcal{H}$. We first observe that the candidate $y$ is the Condorcet winner of the resulting election. We now prove that the Optimal Attack instance $(\mathcal{C}, \mathcal{Q}, k_a, k_d)$ is equivalent to the Clique instance $(\mathcal{G}, k)$.

In the forward direction, let us assume that $\mathcal{U} = \{v_i : i \in \mathcal{I}\} \subset \mathcal{V}$ with $|\mathcal{I}| = k$ forms a clique in $\mathcal{G}$. We claim that attacking all the voter groups $\mathcal{G}_i, i \in \mathcal{I}$ forms a successful attack. Indeed, suppose the defender defends all the voter groups $\mathcal{G}_i, i \in \mathcal{I}$ except $\mathcal{G}_\ell$ and $\mathcal{G}_{\ell'}$. Let $e_{j^\star}$ be the edge between the vertices $v_\ell$ and $v_{\ell'}$ in $\mathcal{G}$. Let the profile after the attack be $\hat{\mathcal{G}}$; that is, $\hat{\mathcal{G}} = \cup_{i \in [n] \setminus \mathcal{I}} \mathcal{G}_i \cup \mathcal{G}_\ell \cup \mathcal{G}_{\ell'} \cup \mathcal{H}$. Then we have $\mathcal{D}_{\hat{\mathcal{G}}}(y, x_{j^\star}) = 0$ and thus the candidate $y$ is not the unique winner after the attack. Hence the Optimal Attack instance is Yes instance.

In the other direction, let the Optimal Attack instance be a Yes instance. We can assume, without loss of generality, that the attacker does not attack the voter group $\mathcal{H}$ since the candidate $y$ loses every pairwise election in $\mathcal{H}$. Let attacking all the voter groups $\mathcal{G}_i, i \in \mathcal{I}$ with $|\mathcal{I}| \leqslant k$ is a successful attack. We observe that if $|\mathcal{I}| < k$, then defending any $k - 2$ of the groups that are attacked foils the attack – since the candidate $y$ continues to be the Condorcet winner in the election, resulting after deleting any one group. Hence we have $|\mathcal{I}| = k$. Let us consider the subset of vertices $\mathcal{U} = \{v_i : i \in \mathcal{I}\}$. We claim that $\mathcal{U}$ forms a clique in $\mathcal{G}$. Indeed, if not, then let us assume that there exists two indices $\ell, \ell' \in \mathcal{I}$ such that there is no edge between the vertices $v_\ell$ and $v_{\ell'}$ in $\mathcal{G}$. Let us consider the defender strategy of defending all the voter groups $\mathcal{G}_i, i \in \mathcal{I} \setminus \{\ell, \ell'\}$. We observe that the candidate $y$ continues to be the Condorcet winner in the resulting election. This contradicts our assumption that attacking all the voter groups $\mathcal{G}_i, i \in \mathcal{I}$ with $|\mathcal{I}| \leqslant k$ is a successful attack. Hence $\mathcal{U}$ forms a clique in $\mathcal{G}$ and thus the Clique instance is a Yes instance. ◀

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 [25]. We observe that the reductions from the Hitting Set problem (parameterized by the size of universe) to the Optimal Defense problem in Theorems 16 and 18 are polynomial parameter transformations. Hence we immediately have the following corollary.

▶ **Corollary 27.** *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 22 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. [69] 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. This result shows that the OPTIMAL DEFENSE problem is fixed parameter tractable with $(k_a, k_d)$ as the parameter.

▶ **Theorem 28.** *There is an algorithm for the* OPTIMAL DEFENSE *problem for every scoring rule and the Condorcet voting rule which runs in time* $\mathcal{O}^*(k_a^{k_d})$.

**Proof.** Let us prove the result for any scoring rule. The proof for the Condorcet voting rule is exactly similar. Let $c$ be the winner of the election. We check if there exist at most $k_a$ groups whose deletion changes the winner. This can be done by iterating over all non-winners $x$ and check if deleting the $k_a$ voter groups where the score of $c$ minus the score of $x$ is maximum makes $c$ not win the election. If no such set of voter groups exist, then we output YES. Otherwise there exists such a set of at most $k_a$ voter groups. We recursively branch on $k_a$ cases by protecting one of these $k_a$ groups in each branch and running the attacking algorithm again. In addition, the parameter $k_d$ is also reduced by 1 each time a group is protected. When $k_d$=0, the attacking algorithm is run on all the leaves of the tree and a valid protection strategy exists as long as for at least one of the leaves the attack outputs no, i.e., after deploying resources to protect $k_d$ groups the attacker is unable to change the outcome of the election with any strategy. The groups to be protected are determined by traversing the tree that leads to the particular leaf which did not output an attack. Clearly the number of nodes in this tree is bounded by $k_a^{k_d}$. The amount of time taken to find an attack at each node is bounded by $\text{poly}(n)$. Hence the running time of this algorithm is bounded by $k_a^{k_d}.\text{poly}(n)$. ◀

## 6 Experiments

Though the previous sections show that the optimal defending problem is computationally intractable, it considers a worst-case instance for the defending problem. It does not say how difficult the defending problem is in real or statistically generated elections. In this section, we conduct an empirical study to understand how simple defending strategies perform for such real or statistically generated elections. The defending strategies we consider are variants of a greedy policy, which are easily computable.

*Defending strategy*: For a given voting profile and a voting rule, the defending strategy finds the set of voter groups to defend. The steps involved are as follows. First, it computes the actual winner, which is, say $a$. The strategy considers $a$ with every other candidate, and for each such ($a$, non-winner) pair, it sorts the groups based on the *winning margin*[1] of votes for $a$ in those groups and picks the top $k_d$ groups to form a list of size $k_d$.

Next, among all these $(m-1)$ sorted lists, each of size $k_d$, the strategy picks the most frequent $k_d$ groups to protect. We call this version of the strategy GREEDY 1. Given a profile, an optimal attacker (a) may be able to change the outcome by attacking some of the unprotected

---

[1] Since we consider only scoring rule based voting schemes in these experiments, the winning margin is the difference in the scores between $a$ and the other candidates.

groups or (b) may be unable to change the outcome. If (a) occurs, then there is a possibility that for the value of $k_d$, there is no defense strategy that can guard the election from all possible attacker strategies. In that case, GREEDY 1 is optimal. GREEDY 1 is not optimal otherwise. However, GREEDY 1 is always optimal for case (b). Note that, given a profile and $k_d$ protected groups, it is easy to find if there exists an optimal attack strategy. At the same time, 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 $k_d$ groups uniformly at random. Call this strategy GREEDY 2.

In this section, we consider two experimental approaches. First, we consider some real election datasets from a standard preferences repository (preflib.org) and run our approaches GREEDY 1 and GREEDY 2 on them. Second, we synthetically generate the preferences using a few standard statistical techniques.

## 6.1 Real election data

We experiment on four real elections: Irish, Minneapolis, Aspen, American Psychological Association Election obtained from the preference repository preflib.org. All four election data are available as voting profiles that consist of complete ordering (with ties) over the candidates. We make the orders *linear* by breaking the ties using an arbitrarily chosen order over the candidates. Then, we partition the votes uniformly at random into 12 groups such that the group sizes are roughly equal. The dataset details are as follows: Irish Dublin West 2002 Election has 9 candidates, and 29988 voters; Minneapolis Board of Estimate and Taxation Election 2009 has 7 candidates, and 32086 voters; Aspen Mayor Election 2009 has 5 candidates, and 2528 voters; and American Psychological Association Election 2009 has 5 candidates and 15313 voters.

We consider three voting rules: plurality, veto, and Borda, and vary $k_d$ between 2 to 10. To study the performance of the GREEDY 1 and GREEDY 2, we keep $k_a = 12 - k_d$ with the assumption that the attacker can attack and destroy all of the undefended voter groups. Quite surprisingly, for each of the four election data, $k_d$, and voting rules $\in \{\text{plurality}, \text{veto}, \text{borda}\}$, the two greedy algorithms successfully defend the election for *all* the 100 random partitions of the votes into 12 groups in this experiment, and therefore are optimal.

A possible reason for this observation is that in those real datasets, the winning margins were significant in all three voting rules. We, therefore, consider the statistically generated voting profiles as explained in the next section to evaluate the performances of the greedy defending rules.

## 6.2 Statistically generated voting profiles

In practice, elections have voting profiles that are generated from some (possibly known) distribution (via surveys or polls). We experiment with three intuitive voter generation models, some of which have also been widely used in the literature: *impartial culture model* ( [26, 41, 45, 63, 64, 66]), *urn model* ( [26, 37, 45, 63, 64]), and *few major candidates model*.

Fix $m = 5$. We generate 1000 preference profiles over these alternatives for $N = 12000$, where each vote is generated from the model. The voters are partitioned into 12 groups containing an equal number of voters.

1. **Impartial Culture (IC) model:** In this model, the votes are drawn uniformly at random from the set of all possible strict preference orders over $m$ alternatives.

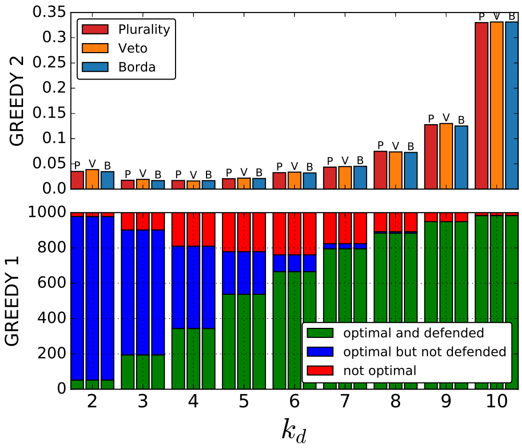


**Figure 1** Performances of GREEDY 1 and GREEDY 2 for IC generation model.

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 $k_d$ groups 100 times. These fractions, therefore, serves as an empirical probability of successful defense of GREEDY 2 given GREEDY 1 is not optimal.

2. **Urn model:** We consider random votes drawn from the Polya-Eggenberger urn model [6]. In the urn model, there is an urn containing all possible $m!$ votes. A vote is drawn uniformly at random from the urn, and is placed back in the urn with $\alpha$ additional copies ($\alpha$ is a non-negative integer). We repeat the procedure N times to generate the voting profile. The correlation between the votes increases with $\alpha$. If $\alpha = 0$, then the *urn model* is identical to the IC model. We analyse the performance of GREEDY 1 and GREEDY 2 for different values of $\alpha$ and found that, as $\alpha$ is increasing, the number of times when the greedy algorithm is optimal and defend the election increases. With $m = 5$, for $\alpha = 30 (= \frac{m!}{4})$ GREEDY 1 is more than 90% times optimal (Figure 2).

3. **Few major candidates model:** Under this model, we consider preferences where the primary contest happens between few major candidates, even though there are more candidates present. E.g., the top candidate for every voter can be one of those few candidates. We first consider two major candidates model, where 40% of the total number of votes have a fixed alternative $a$ on top. Similarly, a different 40% of the total votes have some other alternative $b$ on top. For both these voter groups, the strict order of the rest $(m - 1)$ alternatives is picked uniformly at random. The remaining 20% of the

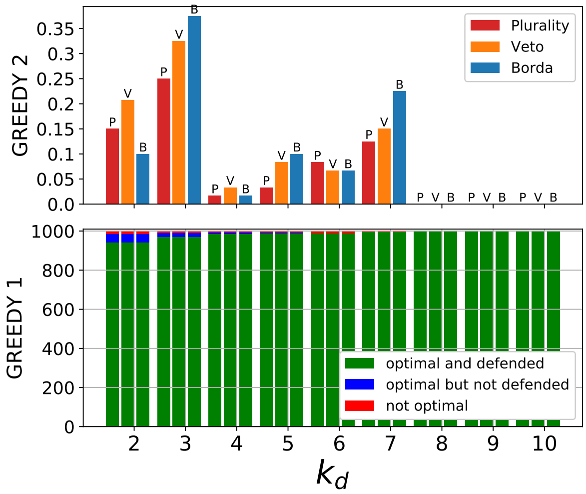


**Figure 2** Performances of GREEDY 1 and GREEDY 2 for voting profiles generated from urn model with $\alpha = 30 = \frac{m!}{4}$.

total votes are picked uniformly at random from the set of all possible strict preference orders over $m$ candidates. Experiments similar to the IC and urn models are run on this generation model and the results are shown in Figure 3. Figure 4 shows a similar plot when there are three major candidates with 25% of the voters keeping them on top and four major candidates with 20% of the voters keeping them on top. The profiles of the rest of the voters and the candidates except the major candidates are similar to the two major candidates model.

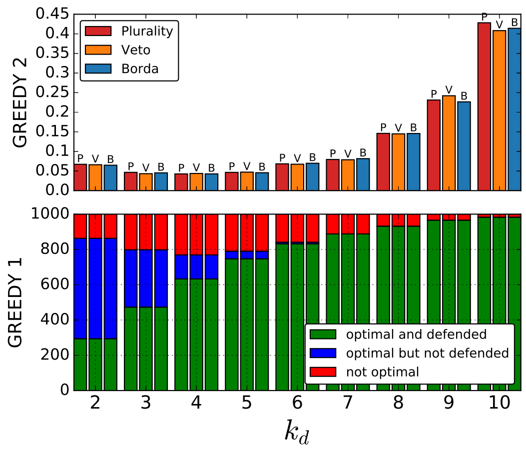


**Figure 3** Performances of GREEDY 1 and GREEDY 2 for voting profile generation model with two major contesting candidates.

The results show that even though optimal defense is a hard problem, a simple strategy like greedy can defend the real elections even with relatively smaller values of $k_d$. For the statistically generated elections, it achieves at least 70% optimality for all generation models. For the rest 30% non-optimal cases, the variant GREEDY 2 is capable of salvaging it into optimal with probability almost 5% for IC model and above 5% for two-major contestant generation model for $k_d = k_a = 6$. This empirically hints at a possibility that defending

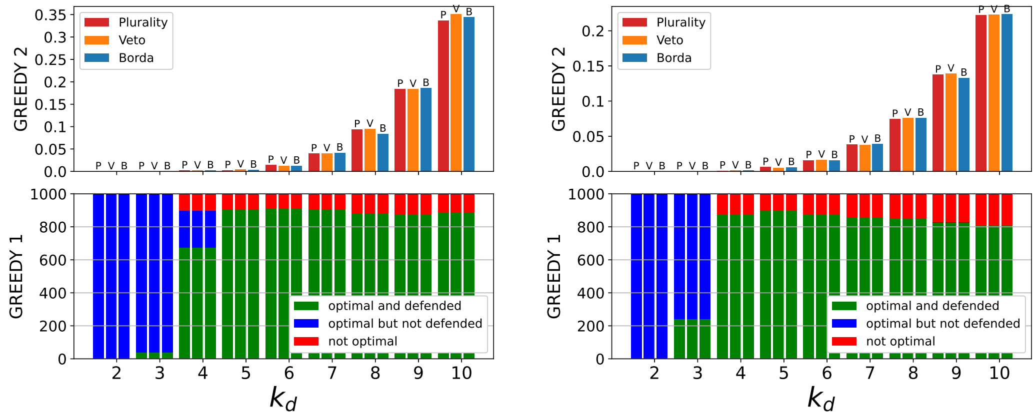


**Figure 4** Performances of GREEDY 1 and GREEDY 2 for voting profile generation model with three (left) and four (right) major contesting candidates.

real-world elections may not be too difficult.

## 7 Conclusion

We have considered the OPTIMAL DEFENSE problem from a primarily parameterized perspective for scoring rules and the Condorcet voting rule. We showed hardness in the number of candidates, the number of resources for the defender or the attacker. On the other hand, we show tractability for the combined parameter $(k_a, k_d)$. We also introduced the OPTIMAL ATTACK problem, which is hard even for the combined parameter $(k_a, k_d)$, and also showed the hardness for a constant number of candidates. Even though the OPTIMAL DEFENSE problem is hard, empirically we show that relatively simple mechanisms ensure good defending performance for reasonable voting profiles.

Several directions for future work emerge from here. First, it will be interesting to see if the running time of the FPT algorithm can be improved by either exploiting structure in the election, or by using some heuristics. An experimental study comparing this approach with the MIP-solvers would be useful to execute. Secondly, all our hardness results exploit the succinct encoding of the problem. It is not clear if the OPTIMAL DEFENSE or OPTIMAL ATTACK problems continue to be hard when the problem is encoded in unary (for a constant number of candidates), or if the problems admit algorithms that are FPT in $m$ at the cost of a pseudo-polynomial overhead in the running time. Thirdly, a fundamental question is if the OPTIMAL ATTACK problem in is coNP. Another interesting question is if there exists any good approximation algorithm for the OPTIMAL ATTACK and OPTIMAL DEFENSE problems. We leave this specific question open.

## References

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

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

3 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.

4 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.

5 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.

6 Sven Berg. Paradox of voting under an urn model: The effect of homogeneity. *Public Choice*, 47(2):377–387, 1985.

7 Nadja Betzler and Johannes Uhlmann. Parameterized complexity of candidate control in elections and related digraph problems. *Theor. Comput. Sci.*, 410(52):5425–5442, 2009.

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

9 Hans L. Bodlaender, Stéphan Thomassé, and Anders Yeo. Kernel Bounds for Disjoint Cycles and Disjoint Paths. In Amos Fiat and Peter Sanders, editors, *Proc. 17th Annual European Symposium,on Algorithms (ESA 2009), Copenhagen, Denmark, September 7-9, 2009.*, volume 5757 of *Lecture Notes in Computer Science*, pages 635–646. Springer, 2009.

10 Felix Brandt, Vincent Conitzer, Ulle Endriss, Jérôme Lang, and Ariel Procaccia. Handbook of computational social choice, 2016.

11 Laurent Bulteau, Jiehua Chen, Piotr Faliszewski, Rolf Niedermeier, and Nimrod Talmon. Combinatorial voter control in elections. *Theor. Comput. Sci.*, 589:99–120, 2015.

12 Jiehua Chen, Piotr Faliszewski, Rolf Niedermeier, and Nimrod Talmon. Elections with few voters: Candidate control can be easy. In *Proc. Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 2045–2051, 2015.

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

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

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

16 Palash Dey. Optimal bribery in voting. In *Proc. International Conference on Autonomous Agents and Multiagent Systems, AAMAS*, 2019.

17 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.

18 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.

19 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.

20 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.

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

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

23 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.

24 Palash Dey, Neeldhara Misra, Swaprava Nath, and Garima Shakya. A parameterized perspective on protecting elections. In Sarit Kraus, editor, *Proc.1 Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 238–244. ijcai.org, 2019.

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

26 Gábor Erdélyi, Michael R Fellows, Jörg Rothe, and Lena Schend. Control complexity in bucklin and fallback voting: An experimental analysis. *Journal of Computer and System Sciences*, 81(4):661–670, 2015.

27 Gábor Erdélyi, Edith Hemaspaandra, and Lane A. Hemaspaandra. More natural models of electoral control by partition. In *Algorithmic Decision Theory - 4th International Conference, ADT 2015, Lexington, KY, USA, September 27-30, 2015, Proceedings*, pages 396–413, 2015.

28 Gábor Erdélyi, Markus Nowak, and Jörg Rothe. Sincere-strategy preference-based approval voting fully resists constructive control and broadly resists destructive control. *Math. Log. Q.*, 55(4):425–443, 2009.

29 Gábor Erdélyi and Jörg Rothe. Control complexity in fallback voting. In *Theory of Computing 2010, CATS 2010, Brisbane, Australia, January 2010*, pages 39–48, 2010.

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

31 Piotr Faliszewski, Edith Hemaspaandra, and Lane A. Hemaspaandra. Weighted electoral control. *J. Artif. Intell. Res. (JAIR)*, 52:507–542, 2015.

32 Piotr Faliszewski, Edith Hemaspaandra, Lane A. Hemaspaandra, and Jörg Rothe. Llull and copeland voting broadly resist bribery and control. In *Proc. Twenty-Second AAAI Conference on Artificial Intelligence, July 22-26, 2007, Vancouver, British Columbia, Canada*, pages 724–730, 2007.

33 Piotr Faliszewski, Edith Hemaspaandra, Lane A. Hemaspaandra, and Jörg Rothe. Copeland voting fully resists constructive control. In *Algorithmic Aspects in Information and Management, 4th International Conference, AAIM 2008, Shanghai, China, June 23-25, 2008. Proceedings*, pages 165–176, 2008.

34 Piotr Faliszewski, Edith Hemaspaandra, Lane A. Hemaspaandra, and Jörg Rothe. Llull and copeland voting computationally resist bribery and constructive control. *J. Artif. Intell. Res. (JAIR)*, 35:275–341, 2009.

35 Piotr Faliszewski, Edith Hemaspaandra, Lane A. Hemaspaandra, and Jörg Rothe. The shield that never was: societies with single-peaked preferences are more open to manipulation and

control. In *Proc. 12th Conference on Theoretical Aspects of Rationality and Knowledge (TARK-2009), Stanford, CA, USA, July 6-8, 2009*, pages 118–127, 2009.

36 Piotr Faliszewski, Edith Hemaspaandra, Lane A. Hemaspaandra, and Jörg Rothe. The shield that never was: Societies with single-peaked preferences are more open to manipulation and control. *Inf. Comput.*, 209(2):89–107, 2011.

37 Piotr Faliszewski, Arkadii Slinko, Kolja Stahl, and Nimrod Talmon. Achieving fully proportional representation by clustering voters. *Journal of Heuristics*, 24(5):725–756, 2018.

38 Zack Fitzsimmons, Edith Hemaspaandra, and Lane A. Hemaspaandra. Control in the presence of manipulators: Cooperative and competitive cases. In *IJCAI 2013, Proc. 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, pages 113–119, 2013.

39 Jörg Flum and Martin Grohe. *Parameterized Complexity Theory*, volume 3. Springer, 2006.

40 Fedor V Fomin, Daniel Lokshtanov, Saket Saurabh, and Meirav Zehavi. *Kernelization: theory of parameterized preprocessing*. Cambridge University Press, 2019.

41 Judy Goldsmith, Jérôme Lang, Nicholas Mattei, and Patrice Perny. Voting with rank dependent scoring rules. In *AAAI*, pages 698–704. Citeseer, 2014.

42 Edith Hemaspaandra, Lane A. Hemaspaandra, and Jörg Rothe. Hybrid elections broaden complexity-theoretic resistance to control. *Math. Log. Q.*, 55(4):397–424, 2009.

43 Edith Hemaspaandra, Lane A. Hemaspaandra, and Jörg Rothe. Controlling candidate-sequential elections. In *ECAI 2012 - 20th European Conference on Artificial Intelligence. Including Prestigious Applications of Artificial Intelligence (PAIS-2012) System Demonstrations Track, Montpellier, France, August 27-31 , 2012*, pages 905–906, 2012.

44 Lane A. Hemaspaandra, Rahman Lavaee, and Curtis Menton. Schulze and ranked-pairs voting are fixed-parameter tractable to bribe, manipulate, and control. In *International conference on Autonomous Agents and Multi-Agent Systems, AAMAS '13, Saint Paul, MN, USA, May 6-10, 2013*, pages 1345–1346, 2013.

45 Orgad Keller, Avinatan Hassidim, and Noam Hazon. New approximations for coalitional manipulation in scoring rules. *Journal of Artificial Intelligence Research*, 64:109–145, 2019.

46 Leon Kellerhals, Viatcheslav Korenwein, Philipp Zschoche, Robert Bredereck, and Jiehua Chen. On the computational complexity of variants of combinatorial voter control in elections. In *Proc 14th International Conference on Theory and Applications of Models of Computation*, pages 348–361. Springer, 2017.

47 Dušan Knop, Martin Koutecký, and Matthias Mnich. A unifying framework for manipulation problems. *arXiv preprint arXiv:1801.09584*, 2018.

48 Dušan Knop, Martin Koutecký, and Matthias Mnich. Voting and bribing in single-exponential time. *ACM Transactions on Economics and Computation (TEAC)*, 8(3):1–28, 2020.

49 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.

50 Hong Liu, Haodi Feng, Daming Zhu, and Junfeng Luan. Parameterized computational complexity of control problems in voting systems. *Theor. Comput. Sci.*, 410(27-29):2746–2753, 2009.

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

52 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.

53 Krzysztof Magiera and Piotr Faliszewski. How hard is control in single-crossing elections? In *ECAI 2014 - 21st European Conference on Artificial Intelligence, 18-22 August 2014, Prague, Czech Republic - Including Prestigious Applications of Intelligent Systems (PAIS 2014)*, pages 579–584, 2014.

54 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.

55 Cynthia Maushagen and Jörg Rothe. Complexity of control by partitioning veto and maximin elections and of control by adding candidates to plurality elections. In *ECAI 2016 - 22nd European Conference on Artificial Intelligence, 29 August-2 September 2016, The Hague, The Netherlands - Including Prestigious Applications of Artificial Intelligence (PAIS 2016)*, pages 277–285, 2016.

56 David C McGarvey. A theorem on the construction of voting paradoxes. *Econometrica*, pages 608–610, 1953.

57 Curtis Menton. Normalized range voting broadly resists control. *Theory Comput. Syst.*, 53(4):507–531, 2013.

58 Curtis Glen Menton and Preetjot Singh. Control complexity of schulze voting. In *IJCAI 2013, Proc. 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, pages 286–292, 2013.

59 Tomasz Miasko and Piotr Faliszewski. The complexity of priced control in elections. *Ann. Math. Artif. Intell.*, 77(3-4):225–250, 2016.

60 Rolf Niedermeier. Invitation to fixed-parameter algorithms. *Habilitationschrift, University of Tübingen*, 2002.

61 David C. Parkes and Lirong Xia. A complexity-of-strategic-behavior comparison between schulze's rule and ranked pairs. In *Proc. Twenty-Sixth AAAI Conference on Artificial Intelligence, July 22-26, 2012, Toronto, Ontario, Canada.*, 2012.

62 Tomasz Put and Piotr Faliszewski. The complexity of voter control and shift bribery under parliament choosing rules. *Trans. Computational Collective Intelligence*, 23:29–50, 2016.

63 Piotr Skowron, Piotr Faliszewski, and Arkadii Slinko. Achieving fully proportional representation: Approximability results. *Artificial Intelligence*, 222:67–103, 2015.

64 Toby Walsh. Where are the hard manipulation problems? *Journal of Artificial Intelligence Research*, 42:1–29, 2011.

65 Jianxin Wang, Weimin Su, Min Yang, Jiong Guo, Qilong Feng, Feng Shi, and Jianer Chen. Parameterized complexity of control and bribery for d-approval elections. *Theor. Comput. Sci.*, 595:82–91, 2015.

66 Jun Wang, Sujoy Sikdar, Tyler Shepherd, Zhibing Zhao, Chunheng Jiang, and Lirong Xia. Practical algorithms for multi-stage voting rules with parallel universes tiebreaking. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 2189–2196, 2019.

67 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.

**68** Lirong Xia and Vincent Conitzer. Determining possible and necessary winners under common voting rules given partial orders. *J. Artif. Intell. Res.*, 41(2):25–67, 2011.

**69** 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.