Protecting Elections by Recounting Ballots

Edith Elkind[1] , **Jiarui Gan**[1] , **Svetlana Obraztsova**[2] ,
Zinovi Rabinovich[2] and **Alexandros A. Voudouris**[1]

[1]Department of Computer Science, University of Oxford
[2]School of Computer Science and Engineering, Nanyang Technological University
{edith.elkind, jiarui.gan, alexandros.voudouris}@cs.ox.ac.uk, {lana, zinovi}@ntu.edu.sg

Abstract

Complexity of voting manipulation is a prominent topic in computational social choice. In this work, we consider a two-stage voting manipulation scenario. First, a malicious party (an attacker) attempts to manipulate the election outcome in favor of a preferred candidate by changing the vote counts in some of the voting districts. Afterwards, another party (a defender), which cares about the voters' wishes, demands a recount in a subset of the manipulated districts, restoring their vote counts to their original values. We investigate the resulting Stackelberg game for the case where votes are aggregated using two variants of the Plurality rule, and obtain an almost complete picture of the complexity landscape, both from the attacker's and from the defender's perspective.

1 Introduction

Democratic societies use elections to select their leaders. However, in societies without a strong democratic tradition, elections may be used as a way to legitimize the status quo: voters are asked to cast their ballots, but the election authorities do not count these ballots correctly, in order to produce an outcome that favors a specific candidate. There are multiple reports of such cases in Russia[1], Congo[2] and Colombia[3], as well as a number of other countries. Even when the election authorities are trustworthy, election results may be corrupted by an external party, for instance, by means of hacking electronic voting machines [Springall *et al.*, 2014; Halderman and Teague, 2015].

There are several ways to counteract electoral fraud. One approach is to send observers to polling stations, to ensure that only eligible voters participate in the election and their ballots are counted correctly. However, it may be infeasible for the party that wants to protect the election (the *defender*) to send observers to all polling stations. Consequently, the election manipulator (the *attacker*) may observe which polling stations remain unprotected, and focus their effort on these stations. Thus, under this approach the attacker benefits from the second-mover advantage.

An alternative approach that the defender can explore is to request recounts in some of the voting districts. While recounts cannot protect from all forms of attacks on election integrity (e.g., a recount is of limited use if voters have been bribed to vote in a specific way, or if the polling station has been burned down), they are feasible in a range of settings and offer the defender the second-mover advantage. Indeed, there are several examples where a recount changed the election outcome. For instance, in the 2008 United States Senate election in Minnesota the Democratic candidate Al Franken won the seat after a recount revealed that 953 absentee ballots were wrongly rejected[4] and in the 2004 race for governor in Washington the Democratic candidate Gregoire was declared the winner after three consecutive recounts[5].

However, recounts can be costly. In Gregoire's case, the Democratic party paid $730000 for a statewide manual recount, and in the 2016 US Presidential Election the fee to initiate a recount in Wisconsin was $3.5 million. Thus, a party that would like to initiate a recount in order to rectify the election results should allocate its budget carefully. Of course, the attacker also incurs costs to carry out the fraud: local election officials may need to be bribed or intimidated, and the more districts are corrupted, the higher is the risk that the election results will not be accepted.

Our Contribution

In this paper we analyze the strategic game associated with vote recounting. In our model, there are two players: the attacker, who modifies some of the votes in order to make his preferred candidate p the election winner, and the defender, who observes the attacker's actions and tries to restore the correct outcome (or, more broadly, to ensure that a candidate better than p wins the election) by means of recounting some of the votes. We assume that the set of voters is partitioned into electoral districts, and both the defender and the attacker make their choices at the level of districts rather than individual votes. The attacker selects a subset of at most $B_\mathcal{A}$ districts and changes the vote counts in the selected districts, and the defender can then restore the vote counts in at most

[1] https://reut.rs/2Gf2FD5

[2] https://on.ft.com/2SW7ggy

[3] https://bit.ly/2V45gDV

[4] https://bit.ly/2S2PMxY

[5] https://bit.ly/2tnO4gG

	Plurality over Voters (PV)	**Plurality over Districts** (PD)	
		Unweighted	Weighted
REC	NP-c, Thm. 3.1 (i) ③ NP-c, Thm. 3.1 (ii) Ⓤ $O(n^{m+2})$, Thm. 3.2	P, Thm. 4.3	NP-c, Thm. 4.1 (i) ③ NP-c, Thm. 4.1 (ii) Ⓤ $O(n^{m+2})$, Thm. 4.2
MAN	NP-h, Thm. 3.3 (i) ③ ⓪ ∞ NP-h, Thm. 3.3 (ii) Ⓤ ⓪ ∞	NP-c, Thm. 4.5 (iii) Ⓤ	Σ_2^P-c, Thm. 4.5 (i) ③ NP-h, Thm. 4.5 (ii) Ⓤ ⓪

Table 1: Summary of our complexity results. MAN denotes the attacker's problem, and REC denotes the defender's problem. Hardness results with Ⓤ hold even when the input is given in unary (the default is binary); with ③ hold even for three candidates; with ⓪ hold even when the defender's budget is zero; with ∞ hold even when the attacker can change as many votes as she wants in each district.

$B_{\mathcal{D}}$ districts to their original values. We assume that both players have full information about the true votes and each other's budgets, and the defender can observe the attacker's actions. While the full information assumption is not entirely realistic, we note that in a district-based model both parties only need to know the vote counts in each district rather than individual votes, and one can get fairly accurate district-level information from independent polls. Also, verifying whether the votes in a district have been tampered with is possible using risk-limiting audits [Lindeman and Stark, 2012; Schürmann, 2016].

For simplicity, we focus on the Plurality voting rule, where each voter votes for a single candidate. We consider two implementations of this rule: (1) Plurality over Voters, where districts are only used for the purpose of collecting the ballots and the winner is selected among the candidates that receive the largest number of votes in total, and (2) Plurality over Districts, where each district selects a preferred candidate using the Plurality rule, and the overall winner is chosen among the candidates supported by the largest number of districts; we also consider a variant of the latter rule where districts have weights, and the measure of a candidate's success is the total weight of districts that support her. Both of these rules are widely used in practice: Plurality over Voters is commonly used in gubernatorial elections in the US, while Plurality over Districts is used in the US Presidential elections.

We provide a detailed analysis of the computational complexity of the algorithmic problems faced by the attacker and the defender. Our main results are summarized in Table 1. Briefly, assuming that the vote counts and the weights of the districts are specified in binary, most of the problems we consider are computationally hard; however, the defender's problem appears to be easier than that of the attacker, and we also get some tractability results for the former. Towards the end of the paper, we consider a variant of our model where the attacker is limited to only transferring support to his preferred candidate; we show that, while this assumption reduces the attacker's ability to achieve his goals, it lowers the complexity of some of the problems we consider.

Related Work

There is a very substantial literature on voting manipulation and bribery; we point the readers to the excellent surveys of Conitzer and Walsh [2016] and Faliszewski and Rothe [2016]. In much of this literature it is assumed that the malicious party can change some of the votes subject to various constraints, and the challenge is to determine whether the attacker's task is computationally feasible; there is no defender that can counteract the attacker's actions. While there is a number of papers that apply game-theoretic analysis to the problem of voting manipulation, they typically consider interactions among several manipulators, with possibly conflicting goals (e.g., see the recent book by Meir [2018]), rather than a manipulator and a socially-minded actor. An important exception, which is similar in spirit to our paper, is the recent work of Yin *et al.* [2018], who investigate a pre-emptive approach to protecting elections. In their model the defender allocates resources to guard some of the electoral districts, so that the votes there cannot be corrupted; notably, in this model the defender has to commit to its strategy first, and the attacker can observe the defender's actions before deciding on its response. The leader-follower (defender-attacker) structure of this model is in the spirit of a series of successful applications of Stackelberg games to security resource allocation problems [Tambe, 2011]. Li *et al.* [2017] analyze a variant of the model of Yin et al. where the goal is to minimize resource consumption, and Chen *et al.* [2018] study a similar scenario, in which manipulation is achieved through bribing the voters. The key difference between our work and the above papers is the action order of the players: in all prior work on election protection that we are aware of the defender makes the first move.

2 The Model

We consider elections over a *candidate set* C, $|C| = m$. There are n *voters* who are partitioned into k pairwise disjoint *districts* $D_1, \dots, D_k$, $k \leq n$; for each $i \in [k]$, let $n_i = |D_i|$. For each $i \in [k]$, district D_i has a *weight* w_i, which is a positive integer; we say that an election is *unweighted* if $w_i = 1$ for all $i \in [k]$. Each voter votes for a single candidate in C. For each $i \in [k]$ and each $a \in C$ let v_{ia} denote the number of votes that candidate a gets from voters in D_i; we refer to the list $\mathbf{v} = (v_{ia})_{i\in[k],a\in C}$ as the *vote profile*.

Let $\succ$ be a linear order over C; $a \succ b$ indicates that a is favored over b. We consider the following two voting rules, which take the vote profile $\mathbf{v}$ as their input.

- *Plurality over Voters* (PV). The winner a^* is chosen from the set $\arg\max_{a\in C} \sum_{i\in[k]} v_{ia}$, with ties broken according to $\succ$. Note that district weights w_i are not relevant for this rule.

- *Plurality over Districts* (PD). For each $i \in [k]$ the winner a_i in D_i is chosen from the set $\arg\max_{a \in C} v_{ia}$, with ties broken according to $\succ$. Then, for each $i \in [k]$, $a \in C$, we set $w_{ia} = w_i$ if $a = a_i$ and $w_{ia} = 0$ otherwise. The winner a^* is chosen from the set $\arg\max_a \sum_{i \in [k]} w_{ia}$, with ties broken according to $\succ$.

For PV and PD, we define the *social welfare* of a candidate $a \in C$ as the total number of votes that a gets and the total weight that a gets, respectively:

$$\mathrm{SW}^{\mathrm{PV}}(a) = \sum_{i \in [k]} v_{ia}, \quad \mathrm{SW}^{\mathrm{PD}}(a) = \sum_{i \in [k]} w_{ia}.$$

Hence, each voting rule selects a candidate that maximizes the respective social welfare.

We consider scenarios where an election may be manipulated by an *attacker*, who wants to change the election result a^* in favor of his preferred candidate $p \in C$. The attacker has a budget $B_{\mathcal{A}} \in [k]$, which means that he can manipulate at most $B_{\mathcal{A}}$ districts. For each $i \in [k]$, we are given an integer γ_i, $0 \leq \gamma_i \leq n_i$, which indicates how many votes the attacker can change in district i if he chooses to manipulate it. Formally, a *manipulation* is described by a set $M \subseteq [k]$, $|M| \leq B_{\mathcal{A}}$, and a vote profile $\tilde{\mathbf{v}} = (\tilde{v}_{ia})_{i \in [k], a \in C}$ such that $\tilde{v}_{ia} = v_{ia}$ for all $i \notin M$, $a \in C$, and for all $i \in [k]$ it holds that $\sum_{a \in C} \tilde{v}_{ia} = n_i$ and $\sum_{a \in C} \max\{0, \tilde{v}_{ia} - v_{ia}\} \leq \gamma_i$.

After the attack, a *defender* with budget $B_{\mathcal{D}} \in \{0\} \cup [k]$ can demand a recount in at most $B_{\mathcal{D}}$ districts. Formally, a defender's strategy is a set $R \subseteq M$ with $|R| \leq B_{\mathcal{D}}$; after the defender acts, the vote counts in all districts in R are restored to their original values, i.e., the resulting vote profile $\mathbf{u} = (u_{ia})_{i \in [k], a \in C}$ satisfies $u_{ia} = v_{ia}$ for each $i \in R$, $a \in C$ and $u_{ia} = \tilde{v}_{ia}$ for each $i \in [k] \setminus R$, $a \in C$. Then the underlying voting rule $\mathcal{R} \in \{\mathrm{PV}, \mathrm{PD}\}$ is applied to $\mathbf{u}$ with ties broken according to $\succ$; let a' denote the candidate selected in this manner. The defender chooses her strategy R so as to maximize $\mathrm{SW}^{\mathcal{R}}(a')$, breaking ties using $\succ$.

We say that the attacker *wins* if he has a strategy $(M, \tilde{\mathbf{v}})$ such that, once the defender responds optimally, candidate p is the winner in the resulting vote profile $\mathbf{u}$; otherwise we say that the attacker *loses*. We note that if $B_{\mathcal{D}} \geq B_{\mathcal{A}}$, the defender can always ensure that $a' = a^*$, i.e., the winner at $\mathbf{u}$ is the winner at the original vote profile $\mathbf{v}$, so in what follows we assume that $B_{\mathcal{D}} < B_{\mathcal{A}}$.

Example 2.1. Consider an election with five districts $D_1, \ldots, D_5$ over a candidate set $C = \{a, b, p\}$, where p is the attacker's preferred candidate; suppose that ties are broken according to the priority order $p \succ a \succ b$. In each of D_1 and D_2 there are 7 voters who vote for a, and in each of D_3, D_4 and D_5 there are 3 voters who vote for b. Suppose that $\gamma_i = n_i$ and $w_i = (n_i)^2$ for each $i \in [5]$, and $B_{\mathcal{A}} = 2$, $B_{\mathcal{D}} = 1$.

If the voting rule is PV, then the attacker does not have a winning strategy. Indeed, consider an attacker's strategy $(M, \tilde{\mathbf{v}})$. If $M \neq \{1, 2\}$, the defender can set $R = M \cap \{1, 2\}$; in the recounted vote profile a gets at least 14 votes, so it is the election winner. If $M = \{1, 2\}$, the defender can set $R = \{1\}$: in the recounted vote profile p gets at most 7 votes, while b gets at least 9 votes, so the winner is a or b (a can win if, e.g., the attacker chooses to transfer exactly 4 votes from a to p in D_2, in which case a gets 10 votes after the recount). Note that even if the winner in $\mathbf{u}$ is b rather than a, the defender still prefers recounting D_1 to no recounting: even though she cannot restore the correct result, she prefers b to p, since $\mathrm{SW}^{\mathrm{PV}}(b) = 9 > 0 = \mathrm{SW}^{\mathrm{PV}}(p)$.

If the voting rule is PD, then the attacker can win by choosing $M = \{1, 2\}$ and transferring the majority of votes from a to p in both manipulated districts. Indeed, even if the defender demands a recount in one of these districts, p still wins the remaining district, leading to a vote weight of 49 in the recounted profile. Since a's vote weight is 49 and b's vote weight is 27, p wins by the tie-breaking rule. □

We assume that both the defender and the attacker have full information about the game. Both parties know the true vote profile $\mathbf{v}$, the parameters w_i and γ_i for each district $i \in [k]$ and each others' budgets. Moreover, the defender observes the strategy $(M, \tilde{\mathbf{v}})$ of the attacker.

We can now define the following decision problems for each $\mathcal{R} \in \{\mathrm{PV}, \mathrm{PD}\}$:

- $\mathcal{R}$-MAN: Given a vote profile $\mathbf{v}$, the attacker's preferred candidate p, budgets $B_{\mathcal{A}}$ and $B_{\mathcal{D}}$, and district parameters $(w_i, \gamma_i)_{i \in [k]}$, does the attacker have a winning strategy?
- $\mathcal{R}$-REC: Given a vote profile $\mathbf{v}$, a distorted vote profile $\tilde{\mathbf{v}}$ with winner b, a candidate $a \neq b$, a budget $B_{\mathcal{D}}$, and district weights $(w_i)_{i \in [k]}$, can the defender recount the votes in at most $B_{\mathcal{D}}$ districts so that a gets elected?

We will also consider an optimization version of $\mathcal{R}$-REC, where a is not part of the input and the goal is to maximize the social welfare of the eventual winner.

Unless specified otherwise, we assume that the vote counts v_{ia} and the district weights w_i are given in binary; we explicitly indicate which of our hardness results still hold if these numbers are given in unary. All problems considered in this paper admit straightforward greedy algorithms for $m = 2$, so in what follows we focus on the case $m \geq 3$. When the voting rule $\mathcal{R} \in \{\mathrm{PV}, \mathrm{PD}\}$ is clear from context, we write $\mathrm{SW}(a)$ instead of $\mathrm{SW}^{\mathcal{R}}(a)$. Due to space constraints, most proofs are omitted.

3 Plurality over Voters

In this section we focus on Plurality over Voters. We first take the perspective of the defender, and then the perspective of the attacker.

Unfortunately, the defender's problem turns out to be computationally hard, even if there are only three candidates or if the input vote counts are given in unary.

Theorem 3.1. PV-REC *is* NP-*complete even when*

(i) $m = 3$, *or*

(ii) the input vote profile is given in unary.

Proof. This problem is clearly in NP. The hardness proof in part (i) follows by a reduction from SUBSET SUM; we omit the details. For part (ii), we give a reduction from EXACT COVER BY 3-SETS (X3C). An instance of this problem is a set E of size 3ℓ and a collection $\mathcal{S}$ of s 3-element subsets of E; it is a yes-instance if there exists a sub-collection $\mathcal{Q} \subseteq \mathcal{S}$ of size ℓ such that $\cup_{S\in\mathcal{Q}} S = E$. Given an instance of X3C, we construct the following PV-REC instance. Without loss of generality, we assume that $\cap_{S\in\mathcal{S}} S = \emptyset$.

- Let $C = \{j_e : e \in E\} \cup \{a, b\}$, $|C| = 3\ell + 2$.
- For each subset $S \in \mathcal{S}$, there is a district D_S with true votes $v_{S,a} = 2$, $v_{S,b} = 6$, $v_{S,j_e} = 2$ for all $e \notin S$, and $v_{S,j_e} = 0$ for all $e \in S$. The attacker distorts the votes in these districts to $\tilde{v}_{S,j} = 2$ for every $j \in C \setminus \{b\}$, $\tilde{v}_{S,b} = 0$.
- There is a district D_{i^*} with true votes $v_{i^*,a} = 6\ell s$, $v_{i^*,b} = 0$, and $v_{i^*,j_e} = 6\ell s + 1$ for every $e \in E$, which is not distorted.
- The budget of the defender is $B_{\mathcal{D}} = \ell$.

Candidate a is the true winner with $2s + 6\ell s$ votes, compared to the $6s$ votes of b and the $2|\{S \in \mathcal{S} : e \notin S\}| + 6\ell s + 1 \leq 2s + 6\ell s - 1$ votes of j_e for every $e \in E$. In the distorted profile $\tilde{\mathbf{v}}$ candidate a gets $2s + 6\ell s$ votes, candidate b gets 0 votes, and each candidate in $C \setminus \{a, b\}$ gets $2s + 6\ell s + 1$ votes. The goal of the defender is to restore candidate a.

Recounting a district D_S reduces by 2 the votes of each candidate j_e such that $e \in S$, leading to a getting more votes than these candidates; b cannot get more than $6s$ votes no matter what the defender does. Therefore, a can be restored as the winner by recounting ℓ districts if and only if E can be covered by ℓ sets from $\mathcal{S}$. □

If the number of candidates is bounded by a constant and the input is given in unary, an optimal set of districts to recount can be identified in time polynomial in the input size by means of dynamic programming.

Theorem 3.2. PV-REC *can be solved in time* $O(k \cdot B_{\mathcal{D}} \cdot n^m)$.

We obtain similar hardness results for the attacker's problem. However, it is not clear if PV-MAN is in NP. Indeed, it may belong to a higher level of the polynomial hierarchy: it is not hard to see that PV-MAN is in Σ_2^P, and it is plausible that this problem is hard for this complexity class.

Theorem 3.3. PV-MAN *is* NP*-hard even when* $B_{\mathcal{D}} = 0$, $\gamma_i = n_i$ *for all* $i \in [k]$ *and*

(i) $m = 3$, *or*

(ii) the input vote profile is given in unary.

In the hardness reductions in the proof of Theorem 3.3 the defender's budget is 0. This indicates that the attacker's problem remains NP-hard even if the defender is known to use a heuristic (e.g., a greedy algorithm) to compute her response.

We remark that PV-REC and PV-MAN with $B_{\mathcal{D}} = 0$ are very similar in spirit to combinatorial (shift) bribery [Bredereck *et al.*, 2016]. In both models, a budget-constrained agent needs to select a set of vote-changing actions, with each action affecting a group of voters. However, there are a few technical differences between the models. For instance, in our model different actions are associated with non-overlapping groups of voters, which is not the case in combinatorial shift bribery. On the other hand, in shift bribery under the Plurality rule votes can only be transferred to/from the manipulator's preferred candidate p, while our model does not impose this constraint (see, however, Section 5). Consequently, it appears that the technical results in our paper cannot be derived from known results for combinatorial shift bribery.

4 Plurality over Districts

In this section we study Plurality over Districts. For the defender's problem, we can replicate the results we obtain for Plurality over Voters, by using similar techniques; in particular, for both of our hardness results it suffices to set $w_i = n_i$ for each $i \in [k]$.

Theorem 4.1. PD-REC *is* NP*-complete even when*

(i) $m = 3$, *or*

(ii) the input vote profile and district weights are given in unary.

Theorem 4.2. PD-REC *can be solved in time* $O(k \cdot B_{\mathcal{D}} \cdot n^m)$.

We also obtain a positive result that does not have an analogue in the PV setting; if all districts have the same weight, the recounting problem can be solved efficiently.

Theorem 4.3. PD-REC *can be solved in polynomial time if* $w_i = 1$ *for all* $i \in [k]$.

Proof. We reduce our problem to nonuniform bribery [Faliszewski, 2008]. An instance of nonuniform bribery under the Plurality rule is given by a set of voters and a set of candidates; for each voter i and each candidate c there is a price p_{ic} for making voter i vote for c, and the briber's goal is to make her preferred candidate the Plurality winner[6] while staying within a budget B. This problem is known to be in P [Faliszewski, 2008]. To reduce PD-REC to nonuniform bribery, we map each district D_i to a single voter i; if the true winner in D_i is x, and in the distorted profile the winner in D_i is y, we set $p_{iy} = 0$, $p_{iz} = +\infty$ for $z \in C \setminus \{x, y\}$, and if $x \neq y$ (i.e., if the attacker has changed the outcome in D_i), we set $p_{ix} = 1$. Then for any candidate $c \in C$ it holds that in PD-REC the defender can make c win by recounting at most $B_{\mathcal{D}}$ districts if and only if in our instance of nonuniform bribery the briber can make c win by spending at most $B_{\mathcal{D}}$. □

We now consider the attacker's problem. Interestingly, for the PD rule, we can actually show a Σ_2^P-hardness result. Our reduction uses a variant of the SUBSET SUM problem, which we term SUB-SUBSET SUM (SSS); this problem may be of independent interest. An instance of this problem is a set $X \subseteq \mathbb{Z}$ and a positive integer ℓ. It is a yes-instance if there is a subset $X' \subseteq X$ with $|X'| = \ell$ such that $\sum_{x\in X''} x \neq 0$ for every non-empty subset $X'' \subseteq X'$. Our proof proceeds by establishing that SSS is Σ_2^P-complete (Lemma 4.4), and then reducing this problem to PD-MAN.

PD-MAN remains computationally hard when the input is given in unary; actually, we conjecture that it remains Σ_2^P-complete, but we were only able to prove NP-hardness. If

[6] Faliszewski [2008] assumes that ties are broken in favor of the briber, but his results extend to lexicographic tie-breaking.

we further assume that all districts have the same weight, the problem can be shown to be NP-complete. These results are summarized in Theorem 4.5.

Lemma 4.4. *SSS is Σ_2^P-complete.*

Theorem 4.5. PD-MAN *is* NP*-hard, and more specifically:*

(i) Σ_2^P*-complete, even when* $m = 3$*;*

(ii) NP*-hard, even when* $B_{\mathcal{D}} = 0$ *and the input vote profile and district weights are given in unary;*

(iii) NP*-complete when* $w_i = 1$ *for all* $i \in [k]$*.*

Proof. We give the proof for part (i). The hardness results in parts (ii) and (iii) follow by reductions from the INDEPENDENT SET problem.

Clearly, PD-MAN is in Σ_2^P. To prove hardness, we reduce from SSS. Given an instance $\langle X, \ell\rangle$ of SSS, we construct an instance of PD-MAN with three candidates $\{a, b, p\}$. Let $X^+ = \{x \in X : x > 0\}$ and $X^- = X \setminus X^+$. Set $y = \sum_{x\in X} 3|x|$. In what follows we describe the votes in each district D_i as a list (v_{ia}, v_{ib}, v_{ip}). The districts are partitioned into three sets I_1, I_2 and I_3:

- I_1 has a district with votes $(0, 3x, 0)$ for each $x \in X^+$, and a district with votes $(0, 0, -3x)$ for each $x \in X^-$.
- I_2 consists of a single district with votes $(0, y + 3, 0)$.
- I_3 consists of three districts with votes $(2y + 5, 0, 0)$, $(0, y - \sum_{x\in X^+} 3x, 0)$, and $(0, 0, 2y + 4 + \sum_{x\in X^-} 3x)$.

For every district D_i we set $w_i = n_i$. The attacker is allowed to change all votes in each district in I_1 and I_2, but none in I_3. Finally, let $B_{\mathcal{A}} = \ell + 1$ and $B_{\mathcal{D}} = \ell$. The true winner in this profile is candidate a with weight $2y + 5$, compared to the weight $2y + 3$ of b and $2y + 4$ of p.

Given a set of integers $Y \subseteq X$, let $I_1(Y)$ be the corresponding set of districts in I_1. Assume that there is a subset $X' \subseteq X$ with $|X'| = \ell$ such that no $X'' \subseteq X'$ has sum equal to 0. The attacker can then exchange the weights of b and p in the districts in $I_1(X')$ and the district in I_2. This way, p becomes the winner with weight $3y + 7 + \sum_{x\in X'} 3x \geq 2y + 7$, compared to the weight $2y + 5$ of a and the weight $y - \sum_{x\in X'} 3x \leq 2y$ of b.

Since SW$(p) >$ SW(b), to defeat the attacker, the defender needs to restore a as the winner. To this end, she must recount the district in I_2, as otherwise p's weight will remain at least $2y + 7$. Hence she can recount at most $\ell - 1$ manipulated districts in I_1. Let the set of non-recounted districts in I_1 be $I_1(X'')$ for some $X'' \subseteq X'$; note that $X'' \neq \varnothing$, so by assumption, $\sum_{x\in X''} x \neq 0$. Then, the weight of b is $2y+3-\sum_{x\in X''} 3x$ and the weight of p is $2y + 4 + \sum_{x\in X''} 3x$. At least one of these numbers is greater than or equal to $2y + 6$; thus, a cannot be restored as the winner.

Conversely, suppose that for every subset $X' \subseteq X$ of size ℓ there exists a non-empty $X'' \subseteq X'$ such that $\sum_{x\in X''} x = 0$. Then, the attacker cannot win. Indeed, let M be the set of manipulated districts. If a district is changed in favor of a, the defender can recount all other districts in M. On the other hand, if all districts in M are won by b or p, the defender can identify a non-empty subset of $M \cap I_1$ such that the corresponding integers sum up to 0, and request a recount of all other districts in M. Such a recount recovers the correct weights of b and p, and a is restored as the winner. □

The second hardness result of Theorem 4.5 holds even for $B_{\mathcal{D}} = 0$, but for the first and the third part of this theorem this is not the case. In fact, PD-MAN is in NP when $B_{\mathcal{D}} = 0$, since the attacker simply needs to guess a manipulation and check whether it makes p the winner. The unweighted case (part (iii)) can be shown to be in P when $B_{\mathcal{D}} = 0$, using a reduction to nonuniform bribery similar to the one in the proof of Theorem 4.3. Thus, recounting has a clear impact on the complexity of the attacker's problem.

5 Regular Manipulations

In our model, the attacker does not have to transfer votes to his preferred candidate p in the manipulated districts; indeed, he may even choose to transfer votes *from* p to another candidate. However, manipulations that give additional votes to candidates other than p are counterintuitive and may be difficult to implement in practice. Therefore, in this section we study what happens if the attacker is limited to transferring votes (in case of PV) or vote weight (in case of PD) to his preferred candidate p.

Definition 5.1 (Regular manipulation)**.** Let p be the preferred candidate of the attacker. A manipulation $(M, \tilde{\mathbf{v}})$ is said to be *regular* if for every district $i \in M$ it holds that

- the voting rule is PV and $\tilde{v}_{ia} \leq v_{ia}$ for all $a \in C \setminus \{p\}$;
- the voting rule is PD and in $\tilde{\mathbf{v}}$ candidate p is the winner in each district in M.

The difference between our general model and the one where the attacker is limited to using regular manipulations is similar to the difference between swap bribery and shift bribery [Elkind *et al.*, 2009]: in swap bribery the attacker can change the vote in any way he likes subject to budget constraints, while in shift bribery he is limited to shifting his preferred candidate in voters' rankings.

One may expect that the restriction to regular manipulations is without loss of generality: indeed, why would the attacker want to transfer votes to candidates other than p? However, our next example shows that this intuition is incorrect.

Example 5.2. We focus on PV; our example also works for PD by setting $w_i = n_i$ for every $i \in [k]$. Consider an instance with 3 candidates $\{a, b, p\}$ and 19 voters who are distributed to 12 districts. The vote profile is as follows:

Candidate	D_1	D_2	$D_3, \dots, D_8$	$D_9, \dots, D_{12}$
a	0	3	1	0
p	6	0	0	0
b	0	0	0	1

Also, $B_{\mathcal{A}} = 2$, $B_{\mathcal{D}} = 1$, and $\gamma_i = n_i$ for all $i \in [12]$. The true winner is candidate a with 9 votes, compared to the 6 votes of p and the 4 votes of b. No regular manipulation can make p win: no matter what the attacker does, by recounting at most one district the defender can ensure that a gets at least 8 votes and p gets at most 7 votes.

Now, consider a non-regular manipulation that distorts all votes in D_1 in favor of b, and all votes in D_2 in favor of p.

Then in the distorted profile a has 6 votes and p has 3 votes, and b wins with 10 votes. If the defender does not recount D_1, b remains the winner after recounting, and if she does recount it, p becomes the winner. Crucially, since $\mathrm{SW}(b) < \mathrm{SW}(p)$, the defender prefers the latter option, so p wins after the recount. □

Example 5.2 shows that only considering regular manipulations may be suboptimal for the attacker. However, the attacker may be limited to regular manipulations by practical considerations. For instance, the election officials in the manipulated districts may find it difficult to follow complex instructions. Thus, it is interesting to understand if focusing on regular manipulations affects the complexity of the problems we consider. It turns out that this is indeed the case.

Let $\mathcal{R} \in \{\mathrm{PV}, \mathrm{PD}\}$, and consider a regular manipulation $(M, \tilde{\mathbf{v}})$. Note that if p is not a winner at $\tilde{\mathbf{v}}$, the attacker necessarily loses: since $(M, \tilde{\mathbf{v}})$ is regular, any recounts can only decrease the votes/weight of p and will not decrease the votes/weight of the current winner. Thus, we can assume that p is the winner at $\tilde{\mathbf{v}}$. The defender can then try the following greedy strategy. Initially, it defines the set of *provisional winners* to consist of p. Then, for each $a \in C \setminus \{p\}$ such that $\mathrm{SW}^{\mathcal{R}}(a) > \mathrm{SW}^{\mathcal{R}}(p)$ or $\mathrm{SW}^{\mathcal{R}}(a) = \mathrm{SW}^{\mathcal{R}}(p)$ and $a \succ p$ the algorithm sorts the districts in M in non-increasing order of the quantity $(v_{ia} - v_{ip}) - (\tilde{v}_{ia} - \tilde{v}_{ip})$ for PV, and the quantity $(w_{ia} - w_{ip}) - (\tilde{w}_{ia} - \tilde{w}_{ip})$ for PD; ties are broken arbitrarily. Next, it checks what happens if the first $B_{\mathcal{D}}$ districts in this order are recounted; if this results in a candidate $b \in C \setminus \{p\}$ with $\mathrm{SW}^{\mathcal{R}}(b) > \mathrm{SW}^{\mathcal{R}}(p)$ or $\mathrm{SW}^{\mathcal{R}}(b) = \mathrm{SW}^{\mathcal{R}}(p)$, $b \succ p$ winning the election, the defender adds b to the set of provisional winners. Finally, it outputs the provisional winner with the maximum social welfare, breaking ties according to $\succ$. We refer to this algorithm as *greedy recounting*; note that its running time is polynomial in the input size.

Lemma 5.3. *Let $\mathcal{R} \in \{\mathrm{PV}, \mathrm{PD}\}$. Suppose that the attacker uses a regular manipulation $(M, \tilde{\mathbf{v}})$. Then greedy recounting outputs p if and only if $(M, \tilde{\mathbf{v}})$ is a winning strategy for the attacker.*

Notably, greedy recounting does not constitute an algorithm for $\mathcal{R}$-REC: it is unable to decide whether there is a recounting strategy that results in a specific candidate becoming the election winner. However, Lemma 5.3 implies that if the attacker is limited to using regular manipulations, his decision problem $\mathcal{R}$-MAN is in NP: he can guess a regular manipulation and use greedy recounting to verify whether it is successful. As the hardness proofs in Theorem 3.3 use regular manipulations, PV-MAN is NP-complete in this case.

Theorem 5.4. *For regular manipulations,* PV-MAN *is* NP-*complete. The hardness result holds even if $m = 3$ or if the input vote profile and district weights are given in unary.*

In contrast, the hardness proofs for PD-MAN (Theorem 4.5) rely on the attacker using a non-regular strategy, and hence they do not imply that PD-MAN remains hard when the attacker is restricted to regular manipulations. In fact, this variant of PD-MAN is in P.

Theorem 5.5. *For regular manipulations,* PD-MAN *can be solved in polynomial time.*

Moreover, it turns out that greedy recounting serves as a $1/2$-approximation algorithm for the defender: it outputs a candidate a such that for every candidate a' that can be made a winner by recounting at most $B_{\mathcal{D}}$ districts it holds that $\mathrm{SW}(a) \geq \mathrm{SW}(a')/2$.

Theorem 5.6. *For regular manipulations, greedy recounting is a $1/2$-approximation algorithm for the optimization versions of* PV-REC *and* PD-REC.

In fact, the bound on the approximation ratio provided by Theorem 5.6 is essentially tight.

Theorem 5.7. *For regular manipulations and any constant $\varepsilon > 0$, neither* PV-REC *nor* PD-REC *admit a polynomial-time $(\frac{1}{2} + \varepsilon)$-approximation algorithm unless* P = NP*, even when $m = 3$.*

6 Conclusion and Open Problems

We have studied the problem of protecting elections by means of recounting votes in the manipulated districts. Our results offer an almost complete picture of the worst-case complexity of the problems faced by the defender and the attacker. Perhaps the most obvious open question is whether we can strengthen the NP-hardness results for PV-MAN and for PD-MAN under unary representation to Σ_2^P-completeness results. The next challenge is to extend our results beyond Plurality; e.g., leadership elections are often conducted using Plurality with Runoff, and it would be interesting to understand if similar results hold for this rule.

Our model is quite expressive: districts may have different weights, and an attacker may only be able to corrupt a fraction of votes in a district. These features of the model are intended to capture the challenges of real-world scenarios; in particular, it is typically infeasible for the attacker to change *all* votes in a district. However, it is important to understand their impact on the complexity of the problems we consider. We tried to indicate which of our hardness results hold for special cases of the model, and proved some easiness results under simplifying assumptions, but it would be good to obtain a more detailed picture, possibly using the tools of parameterized complexity. A concrete open question is whether our Σ_2^P-hardness result holds if $\gamma_i = n_i$ for all $i \in [k]$.

We contrasted out model with that of Yin *et al.* [2018], where the defender moves first and protects some of the districts from manipulation. In practice, the defender can use a variety of protective measures at different points in time, and an exciting direction for future work is to analyze what happens when the defender can split her resources among different activities, with some activities preceding the attack, and others (such as recounting) undertaken in the aftermath of the attack.

Acknowledgements

This work is supported by the European Research Council (ERC) under grant number 639945 (ACCORD), by the EPSRC International Doctoral Scholars Grant EP/N509711/1, by the MOE AcRF-T1-RG23/18 grant, and by the NTU SUG M4081985 grant.

References

[Bredereck *et al.*, 2016] Robert Bredereck, Piotr Faliszewski, Rolf Niedermeier, and Nimrod Talmon. Large-scale election campaigns: Combinatorial shift bribery. *Journal of Artificial Intelligence Research*, 55:603–652, 2016.

[Chen *et al.*, 2018] Lin Chen, Lei Xu, Shouhuai Xu, Zhimin Gao, Nolan Shah, Yang Lu, and Weidong Shi. Protecting election from bribery: New approach and computational complexity characterization. In *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1894–1896, 2018.

[Conitzer and Walsh, 2016] Vincent Conitzer and Toby Walsh. Barriers to manipulation in voting. In *Handbook of Computational Social Choice*, pages 127–145. 2016.

[Elkind *et al.*, 2009] Edith Elkind, Piotr Faliszewski, and Arkadii Slinko. Swap bribery. In *Proceedings of the 2nd International Symposium on Algorithmic Game Theory (SAGT)*, pages 299–310, 2009.

[Faliszewski and Rothe, 2016] Piotr Faliszewski and Jörg Rothe. Control and bribery in voting. In *Handbook of Computational Social Choice*, pages 146–168. 2016.

[Faliszewski, 2008] Piotr Faliszewski. Nonuniform bribery. In *Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1569–1572, 2008.

[Halderman and Teague, 2015] J. Alex Halderman and Vanessa Teague. The New South Wales iVote system: Security failures and verification flaws in a live online election. In *Proceedings of the 5th International Conference on E-Voting and Identity (VoteID)*, pages 35–53, 2015.

[Li *et al.*, 2017] Yunpeng Li, Yichuan Jiang, and Weiwei Wu. Protecting elections with minimal resource consumption. In *Proceedings of the 16th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1595–1597, 2017.

[Lindeman and Stark, 2012] Mark Lindeman and Philip B. Stark. A gentle introduction to risk-limiting audits. *IEEE Security & Privacy*, 10(5):42–49, 2012.

[Meir, 2018] Reshef Meir. *Strategic Voting*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2018.

[Schürmann, 2016] Carsten Schürmann. A risk-limiting audit in Denmark: A pilot. In *Proceedings of the 1st International Joint Conference on Electronic Voting (E-Vote-ID)*, pages 192–202, 2016.

[Springall *et al.*, 2014] Drew Springall, Travis Finkenauer, Zakir Durumeric, Jason Kitcat, Harri Hursti, Margaret MacAlpine, and J. Alex Halderman. Security analysis of the Estonian internet voting system. In *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS)*, pages 703–715, 2014.

[Tambe, 2011] Milind Tambe. *Security and game theory: algorithms, deployed systems, lessons learned*. Cambridge University Press, 2011.

[Yin *et al.*, 2018] Yue Yin, Yevgeniy Vorobeychik, Bo An, and Noam Hazon. Optimal defense against election control by deleting voter groups. *Artificial Intelligence*, 259:32–51, 2018.