Protecting Elections by Recounting Ballots*

Edith Elkind[1], Jiarui Gan[1], Svetlana Obraztsova[2],
Zinovi Rabinovich[2] and Alexandros A. Voudouris[3]

[1]Department of Computer Science, University of Oxford
[2]School of Computer Science and Engineering, Nanyang Technological University
[3]School of Computer Science and Electronic Engineering, University of Essex

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 elections and their ballots are counted correctly. However, it may be infeasible for the party that wants to protect the elections (the *defender*) to send observers to all polling stations. Consequently, the election manipulator (the *attacker*) may find out which polling stations remain unprotected, and focus their effort on these stations. Thus, under this approach the attacker benefits from the second-mover advantage.

*A preliminary version of this paper appeared in the *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, 2019 [Elkind et al., 2019]. This full version includes all proofs that were omitted from the conference version as well as additional examples and algorithmic results, such as a pseudo-polynomial time algorithm for the weighted version of the recounting problem for Plurality over Districts when the attacker is limited to regular manipulations, and a polynomial time algorithm for the unweighted version of the recounting problem for Plurality over Districts under an additional technical assumption. This work has been supported by the ERC Starting Grant 639945 (ACCORD), the EPSRC International Doctoral Scholars Grant EP/N509711/1, MOE AcRF-T1-RG23/18 grant, and the NTU SUG M4081985 grant.

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

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

[3]https://colombiareports.com/voting-fraud-in-colombia-how-elections-are-rigged/

	Plurality over Voters (PV)	**Plurality over Districts** (PD)	
		Unweighted	Weighted
Rec	NP-c, Thm. 4.1 (i) ③ NP-c, Thm. 4.1 (ii) Ⓤ $O(n^{m+2})$, Thm. 4.2	P, Thm. 5.3	NP-c, Thm. 5.1 (i) ③ NP-c, Thm. 5.1 (ii) Ⓤ $O(n^2 \cdot W^m)$, Thm. 5.2
Rec-Reg	NP-c, Thm. 6.6 ③ NP-c, Thm. 6.7 Ⓤ $\frac{1}{2}$-approximable, Thm. 6.5		NP-c, Thm. 6.6 ③ $O(m \cdot k^2 \cdot W + m \cdot k^3)$, Thm. 6.8 $\frac{1}{2}$-approximable, Thm. 6.5
Man	NP-h, Thm. 4.3 (i) ③ ⓪ (∞) NP-h, Thm. 4.3 (ii) Ⓤ ⓪ (∞)	NP-c, Thm. 5.8 Ⓤ P, Cor. 6.13 (∞)* P, Prop. 5.10 ⓪	Σ_2^P-c, Thm. 5.6 ③ NP-h, Thm. 5.7 Ⓤ ⓪
Man-Reg	NP-c, Thm. 6.9 ③ ⓪ (∞) NP-c, Thm. 6.9 Ⓤ ⓪ (∞)		P, Thm. 6.10

Table 1: Summary of our complexity results. Man denotes the attacker's problem, and Rec denotes the defender's problem; Man-Reg and Rec-Reg denote the versions of these problems where the manipulator can only boost the vote count of his preferred candidate; see Sections 2 and 6 for the formal definitions of these problems. The variables n, m, k and W denote, respectively, the number of voters, the number of candidates, the number of districts and the sum of the weights of all districts. Hardness results marked by Ⓤ hold even when the input is given in unary (the default is binary); those marked by ③ hold even for three candidates; those marked by ⓪ hold even when the defender's budget is zero; and those marked by (∞) hold even when the attacker can change as many votes as he wants in each district. Similarly, easiness results marked with (∞)/Ⓤ/⓪ hold as long as the respective restriction is satisfied; the easiness result marked with * requires an additional technical assumption on the attacker's budget. We omit the results that are implied by other results in the table.

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.

1.1 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

[4]https://bit.ly/2S2PMxY
[5]https://bit.ly/2tnO4gG

p the election winner, and the defender, who observes the attacker's actions and tries to restore the correct outcome (or, more broadly, to achieve the best possible outcome from the voters' perspective, given the attacker's actions) 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 $B_{\mathcal{D}}$ districts to their original values. To capture the fact that not all districts are equally easy to manipulate, we assume that for each district the attacker is given a bound on the number of votes that he is able to change: e.g., some districts may be non-manipulable, while in other districts the votes may be changed arbitrarily. Both players are assumed to 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 players only need to know the aggregate 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 the districts that support it. Both of these rules are widely used in practice. For example, Plurality over Voters is commonly used in gubernatorial elections in the US, while Plurality over Districts is used in the US Presidential elections.

Our main results are summarized in Table 1. In the first part of the paper, after briefly discussing the easy special case of two candidates (Section 3), we provide a detailed analysis of the computational complexity of the algorithmic problems faced by the attacker and the defender (Sections 4 and 5). Briefly, most of the problems we consider are computationally hard, even if (a) the vote counts and the weights of the districts are specified in unary or (b) the number of candidates is small; however, we obtain easiness results if both of the conditions (a) and (b) are satisfied. Many of our hardness results hold even if the attacker can make arbitrary changes to the votes in the manipulated districts, and the attacker's problem remains hard even if the defender's budget is 0. However, the defender's problem appears to be easier than that of the attacker, and in particular we obtain an efficient algorithm for the defender's problem under Plurality over Districts assuming that all districts have unit weights. The most technically challenging result in this part of the paper is the proof that for Plurality over Districts the attacker's problem is Σ_2^P-complete; the argument proceeds by a sequence of reductions, starting from a 2-player game based on a variant of the Partition problem.

In the second part of the paper (Section 6), we consider a variant of our model where the attacker is limited to only transferring votes/districts to his preferred candidate; we refer to such manipulations as *regular*. It turns out that this constraint reduces the attacker's ability to achieve his goals. However, it also lowers the complexity of some of the problems we consider. Most notably, for Plurality over Districts the problem of finding a successful regular manipulation turns out to be polynomial-time solvable. Intriguingly, finding an optimal recounting strategy in this setting remains computationally hard, so, counterintuitively, the attacker's problem turns out to be easier than the defender's problem, whereas the opposite is true in the general setting, subject to standard complexity assumptions.

Finally, we identify a setting where constraining the attacker to regular manipulations does not restrict his ability to succeed: specifically, we show that if all districts have the same weight, the attacker is able to change the winner in each district, and the attacker's budget is not too large, under

Plurality over Districts an attacker has a winning manipulation if and only if he has a winning regular manipulation. Thus, our analysis of regular manipulations enables us to derive tractability results for the general setting.

1.2 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. Despite this crucial difference, there are several special cases of the problems we consider, especially for Plurality over Districts, that can be naturally phrased as bribery problems; indeed, our work offers new results on the complexity of several special cases of nonuniform bribery under the Plurality rule [Faliszewski, 2008], such as succinct bribery and weighted bribery (as presented in Lemma 5.9 and Theorem 6.8). We give a brief summary of these results in Section 2, after we present the formal definition of our model.

While there is a number of papers that apply game-theoretic analysis to the problem of voting manipulation, they typically consider interactions between 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, Chen et al. [2018] study a similar scenario, in which manipulation is achieved through bribing the voters, and Dey et al. [2019] consider this setting from the parameterized complexity perspective. 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* $D_1, \dots, D_k$, $k \le n$; 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). We say that a candidate a *beats* a candidate b under PV if $\sum_{i \in [k]} v_{ia} > \sum_{i \in [k]} v_{ib}$ or $\sum_{i \in [k]} v_{ia} = \sum_{i \in [k]} v_{ib}$ and $a \succ b$; the winner is the candidate that beats all other candidates. 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 district 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. We say that a candidate a *beats* a candidate b under PD if

$\sum_{i\in[k]} w_{ia} > \sum_{i\in[k]} w_{ib}$ or $\sum_{i\in[k]} w_{ia} = \sum_{i\in[k]} w_{ib}$ and $a \succ b$; the winner is the candidate that beats all other candidates.

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, the winner under each voting rule is a candidate with the maximum 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, $0 \le \gamma_i \le n_i$, 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 $M \subseteq [k]$, $|M| \le B_{\mathcal{A}}$, 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 $\sum_{a\in C} \max\{0, \tilde{v}_{ia} - v_{ia}\} \le \gamma_i$.

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 $|R| \le B_{\mathcal{D}}$; 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 $\succ$. The defender chooses her strategy R so as to maximize the social welfare $\mathrm{SW}^{\mathcal{R}}(a')$ of the candidate a' that is selected in this manner, 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 $B_{\mathcal{D}} \ge B_{\mathcal{A}}$, 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 the attacker's strategy satisfies $|M| > B_{\mathcal{D}}$.

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 $M \ne \{1, 2\}$, 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 not 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 a majority of votes from a to p in both 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 winner after the recount.

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. When the voting rule $\mathcal{R} \in \{\text{PV}, \text{PD}\}$ is clear from the context, we write $\text{SW}(a)$ instead of $\text{SW}^{\mathcal{R}}(a)$.

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 generally impose this constraint, (see, however, the variant of our model considered in Section 6). Consequently, it appears that the technical results in our paper cannot be derived from known results for combinatorial shift bribery and vice versa.

Another well-studied problem in computational social choice that shares many similarities with the problems we consider is 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 π_{ic} for making voter i vote for c, and the briber's goal is to make his preferred candidate the Plurality winner[6] while staying within a budget B. This problem is known to be in P [Faliszewski, 2008]. In what follows, we obtain easiness results for PD-REC with unit weights and for PD-MAN with unit weights and zero budget, by reducing these problems to Plurality nonuniform bribery (Theorem 5.3 and Proposition 5.10, respectively). Also, we prove an auxiliary lemma (Lemma 5.9) that can be interpreted as an easiness result for Plurality nonuniform bribery with unit prices when the input is represented succinctly. Finally, we formulate a weighted version of the Plurality nonuniform bribery problem and describe a polynomial-time algorithm for it (Theorem 6.8).

Next, we give formal definitions of the decision problems that are used throughout the paper to show hardness of $\mathcal{R}$-REC and $\mathcal{R}$-MAN for $\mathcal{R} \in \{\text{PV}, \text{PD}\}$, under various constraints.

Definition 2.2 (EXACT COVER BY 3-SETS (X3C))**.** An instance of X3C is given by a set E of size 3ℓ and a collection $\mathcal{S}$ of 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 $\cup_{S\in\mathcal{Q}} S = E$, and a no-instance otherwise.

Definition 2.3 (INDEPENDENT SET)**.** An instance of INDEPENDENT SET is a graph $G = (V, E)$ and an integer ℓ. It is a yes-instance if there exists a subset $V' \subseteq V$ of size ℓ that forms an independent set, i.e., $\{a, b\} \notin E$ for all $a, b \in V'$, and a no-instance otherwise.

Definition 2.4 (SUBSET SUM)**.** An instance of SUBSET SUM is given by a multiset X of integers. It is a yes-instance if there exists a non-empty subset $X' \subseteq X$ such that $\sum_{x\in X'} x = 0$, and a no-instance otherwise.

Definition 2.5 (PARTITION)**.** An instance of PARTITION is given by a multiset X of positive integers. It is a yes-instance if there exists a subset $X' \subseteq X$ such that $\sum_{x\in X'} x = \frac{1}{2}\sum_{x\in X} x$, and a no-instance otherwise.

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

All of these problems are NP-complete [Garey and Johnson, 1979]. However, Subset Sum and Partition are NP-hard only when the input is given in binary; that is, they admit algorithms whose running time is polynomial in $|X| \times \max_{x \in X} |x|$ and therefore these problems can be solved in time polynomial in the input size when the input is given in unary [Garey and Johnson, 1979, Vazirani, 2001, Kellerer et al., 2004].

3 Warm-up: Two Candidates

We start by considering the complexity of our problem for the case $m = 2$, i.e., when there are only two candidates. We will argue that all problems considered in this paper become polynomial-time solvable under this assumption. This analysis provides useful intuition about the different variants of our model and also establishes that the hardness results in subsequent sections are tight with respect to the number of candidates. Throughout this section, we assume that $C = \{p, a\}$, where p is the attacker's preferred candidate.

We first consider the complexity of the defender's problem.

Proposition 3.1. *Both* PV-Rec *and* PD-Rec *are polynomial-time solvable if* $m = 2$.

Proof. We consider PV-Rec first. The defender can order the districts by the number of votes that were moved from a to p, breaking ties arbitrarily; that is, if $\delta_i = \tilde{v}_{ip} - v_{ip} > \tilde{v}_{jp} - v_{jp}$ then D_i should precede D_j in this order. Let $z = |\{D_i : \delta_i \geq 0\}|$. The defender can then demand a recount in the first $\min\{z, B_\mathcal{D}\}$ districts in this order. This strategy maximizes the number of votes that a can gain as a result of the recount; thus, the defender wins if and only if this strategy makes a a winner under PV.

For PD-Rec we use a similar approach. Let z' be the number of districts where the attacker has changed the winner from a to p. The defender can order these districts by weight from the largest to the smallest, breaking ties arbitrarily, and demand a recount in the first $\min\{z', B_\mathcal{D}\}$ districts in this order. This strategy maximizes the weight that a can gain as a result of the recount. □

Given that it is optimal for the defender to use a greedy recounting strategy, we can show that it is optimal for the attacker to use a greedy strategy as well.

Proposition 3.2. *Both* PV-Man *and* PD-Man *are polynomial-time solvable if* $m = 2$.

Proof. Just as in the proof of Proposition 3.1, the attacker needs to order the districts according to a certain parameter and modify the votes in the first $B_\mathcal{A}$ districts in this order.

For PV, the relevant parameter is $x_i = \min\{\gamma_i, v_{ia}\}$; this is the number of votes the attacker can change from a to p in D_i. Thus, the attacker should order the districts so that D_i precedes D_j whenever $x_i > x_j$, and attack the first $B_\mathcal{A}$ districts in this order, by changing x_i votes in D_i from a to p. The defender will then recount the first $B_\mathcal{D}$ of these districts; it is straightforward to see that if this manipulation strategy fails, so would any other manipulation strategy.

For PD, the attacker needs to identify the districts where he can change the outcome from a to p, order these districts by weight from the largest to the smallest (breaking ties arbitrarily), and attack the first $B_\mathcal{A}$ districts in this list (or all of them, if the number of such districts is less than $B_\mathcal{A}$). Again, the defender will then recount the first $B_\mathcal{D}$ of the manipulated districts, and it is easy to see that if this manipulation strategy fails, so would any other manipulation strategy. □

4 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 4.1. 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. We give separate hardness proofs for the case $m = 3$ (part (i)) and for the case where the input is given in unary (part (ii)); we note that the problem becomes polynomial-time solvable if both of the constraints (i) and (ii) are satisfied (Theorem 4.2).

Part (i). To prove that PV-Rec is NP-hard for $m = 3$, we provide a reduction from Subset Sum; see Definition 2.4.

Given an instance X of Subset Sum with $|X| = \ell$, we construct an instance of PV-Rec as follows. Without loss of generality, we assume that $x \neq 0$ for every $x \in X$ and $\sum_{x \in X} x > 0$, and let $X^+ = \{x \in X : x > 0\}$, $X^- = \{x \in X : x < 0\}$, $y = \sum_{x \in X} 2|x|$. We set $C = \{a, b, p\}$, where p is the attacker's preferred candidate. In what follows, we describe each district D_i by a tuple (v_{ia}, v_{ib}, v_{ip}). There are $n = 3y + 1$ voters distributed over $\ell + 3$ districts, which are further partitioned into two sets I_1 and I_2 as follows:

- For each $x \in X^+$ there is a district in I_1 with votes $(0, 2x, 0)$, which are distorted to $(0, 0, 2x)$, and for each $x \in X^-$ there is a district in I_1 with votes $(0, 0, -2x)$, which are distorted to $(0, -2x, 0)$. Note that $|I_1| = \ell$.
- I_2 contains three districts with votes $(y+1, 0, 0)$, $(0, y - \sum_{x \in X^+} 2x, 0)$, and $(0, 0, y + \sum_{x \in X^-} 2x)$, respectively. The votes in these districts are not distorted.

Finally, $B_{\mathcal{D}} = \ell - 1$.

Before the manipulation, a gets $y + 1$ votes and b and p get y votes each. After the manipulation, a gets $y + 1$ votes, b gets $y - \sum_{x \in X} 2x$ and p gets $y + \sum_{x \in X} 2x$ votes; thus, by our assumption that $\sum_{x \in X} x > 0$, candidate p is the winner in the manipulated profile. The goal is to restore the true winner a.

Assume that there exists a non-empty subset $X' \subseteq X$ such that $\sum_{x \in X'} x = 0$. Then, by recounting the $\ell - |X'|$ districts of I_1 that correspond to the integers in $X \setminus X'$, the defender can ensure that both b and p get y votes. Since a always gets $y + 1$ votes from the non-manipulated districts, it is successfully restored as the winner.

Conversely, assume that there is no non-empty subset $X' \subseteq X$ such that $\sum_{x \in X'} x = 0$. Then, since the votes of b and p always add up to exactly $2y$, and each of them gets an even number of votes from each district, one of them must get at least $y + 2$ votes. Therefore, a cannot be restored as the winner.

Part (ii). We give a reduction from Exact Cover By 3-Sets (X3C); see Definition 2.2. Given an instance of X3C, we construct the following PV-Rec instance. Without loss of generality, we assume that $\cup_{S \in \mathcal{S}} S = E$, and let $s = |\mathcal{S}|$.

- Let $C = \{j_e : e \in E\} \cup \{a, b\}$. Note that $|C| = 3\ell + 2$.
- For each subset $S \in \mathcal{S}$, there is a district D_S, where a gets 2 votes, b gets 6 votes, for each $e \notin S$ candidate j_e gets 2 votes, and for each $e \in S$ candidate j_e gets 0 votes. The attacker distorts the votes in each D_S, $S \in \mathcal{S}$, by transferring two votes from b to each candidate j_e with $e \in S$, so that in the distorted profile in each district D_S, $S \in \mathcal{S}$, b gets 0 votes and every other candidate gets 2 votes.

- There is a district D_0 where a receives $6\ell s$ votes, b receives 0 votes and for every $e \in E$ candidate j_e receives $6\ell s + 1$ votes; the votes in this district are 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.

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 4.2. PV-Rec *can be solved in time* $O(k \cdot B_{\mathcal{D}} \cdot (n+1)^m)$.

Proof. Consider an instance of PV-Rec with a candidate set C, $|C| = m$, and n voters that are distributed over k districts. For each $i \in [k]$, let $\mathbf{v}_i = (v_{ia})_{a \in C}$ and $\tilde{\mathbf{v}}_i = (\tilde{v}_{ia})_{a \in C}$ denote, respectively, the true and distorted votes in district i. Let $B_{\mathcal{D}}$ be the budget of the defender.

We present a dynamic programming algorithm that given a candidate $c \in C$, decides whether c can be made the election winner by recounting at most $B_{\mathcal{D}}$ districts. Our algorithm fills out a table T containing entries of the form $T(i, \ell, \mathbf{u})$, for each $i \in \{0, 1, \ldots, k\}$, $\ell \in \{0, 1, \ldots, B_{\mathcal{D}}\}$, and $\mathbf{u} = (u_a)_{a \in C} \in \{0, \ldots, n\}^m$; thus, $|T| = (k+1) \cdot (B_{\mathcal{D}}+1) \cdot (n+1)^m$. We define $T(i, \ell, \mathbf{u}) =$ *true* if we can recount at most ℓ of the first i districts so that the vote count of candidate a equals u_a for each $a \in C$; otherwise we define $T(i, \ell, \mathbf{u}) =$ *false*. There exists a recounting strategy that makes c the eventual winner if and only if there exists a $\mathbf{u}$ such that $T(k, B_{\mathcal{D}}, \mathbf{u}) =$ *true*, $u_c \geq u_a$ for all $a \in C$, and for all $a \in C \setminus \{c\}$ such that $u_c = u_a$ the tie-breaking rule favors c over a.

For each $a \in C$, let $\tilde{u}_a = \sum_{i \in [k]} \tilde{v}_{ia}$ be the number of votes that candidate a gets after manipulation, and let $\tilde{\mathbf{u}} = (\tilde{u}_a)_{a \in C}$. We fill out T according to the following rule:

$$T(i, \ell, \mathbf{u}) = \begin{cases} \textit{true}, & \text{if } \mathbf{u} = \tilde{\mathbf{u}} \\ \textit{false}, & \text{if } i = 0 \text{ or } \ell = 0, \text{ and } \mathbf{u} \neq \tilde{\mathbf{u}} \\ T(i-1, \ell, \mathbf{u}) \vee (\mathbf{u} - \mathbf{v}_i + \tilde{\mathbf{v}}_i \in \{0, \ldots, n\}^m \text{ and} \\ \qquad T(i-1, \ell-1, \mathbf{u} - \mathbf{v}_i + \tilde{\mathbf{v}}_i) = \textit{true}), & \text{otherwise.} \end{cases}$$

This completes the proof. □

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 4.3. 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.

Proof. We prove the two claims separately.

Part (i). To prove that PV-Man is NP-hard for $m = 3$, we provide a reduction from Subset Sum; see Definition 2.4.

Given an instance X of Subset Sum with $|X| = \ell$, we construct an instance of PV-Man as follows. We can assume without loss of generality that $\ell \geq 2$ and $x \neq 0$ for every $x \in X$, and let $y = \max_{x \in X} 2|x|$; by our assumptions, $y \geq 2$. We set $C = \{a, b, p\}$, where p is the attacker's preferred candidate. In what follows, we describe each district D_i by a tuple (v_{ia}, v_{ib}, v_{ip}). There are $n = 12y\ell + 1$ voters distributed over $k = 4\ell + 2$ districts, which are further partitioned into four sets I_1, I_2, I_3, I_4 as follows:

- For each $x \in X$ there is a district in I_1 with votes $(2y + 4x, 2y - 4x, 0)$. Thus, $|I_1| = \ell$.
- Set I_2 consists of $\ell - 1$ districts with votes $(2y, 2y, 0)$ in each district.
- For each $x \in X$ there are two districts in I_3 with votes $(y - 2x, y + 2x, 0)$. Thus, $|I_3| = 2\ell$.
- Set I_4 consists of three districts with votes $(y, y, 0)$, $(y, y, 0)$, and $(0, 0, 1)$.

We set $B_{\mathcal{A}} = \ell$, $B_{\mathcal{D}} = 0$ and $\gamma_i = n_i$ for each $i \in [k]$.

We have $\mathrm{SW}(a) = \mathrm{SW}(b) = 6y\ell$ and $\mathrm{SW}(p) = 1$. Hence, the true winner is a or b, depending on the tie-breaking rule. We claim that the attacker can make p the winner if and only if there exists a non-empty subset $X' \subseteq X$ such that $\sum_{x \in X'} x = 0$.

To see this, assume first that there exists a subset $X' \subseteq X$ such that $|X'| \geq 1$ and $\sum_{x \in X'} x = 0$. Then the attacker can distort the votes in the $|X'|$ districts of I_1 corresponding to the elements of X', and in arbitrary $\ell - |X'|$ districts of I_2, by transferring all votes to p in each of these districts. In the resulting election, p gets $4y\ell + 1$ votes, while a and b get $4y\ell$ votes each, so p becomes the winner.

Conversely, suppose that the attacker has a successful manipulation $(M, \tilde{\mathbf{v}})$ with $|M| \leq \ell$. For each $c \in C$, let s_c denote the number of votes that c receives in $\tilde{\mathbf{v}}$. For p to be the winner in $\tilde{\mathbf{v}}$, it must hold that $s_p \geq n/3 = (12y\ell + 1)/3$; since s_p is an integer and $\mathrm{SW}(p) = 1$, this means that the manipulation transfers at least $4y\ell$ votes to p. On the other hand, in every district there are at most $4y$ voters who vote for a or b, so p can gain at most $4y\ell$ votes from the manipulation. It follows that $s_p = 4y\ell + 1$, $s_a + s_b = 8y\ell$. Further, $s_p = 4y\ell + 1$, $|M| = \ell$ implies that $M \subseteq I_1 \cup I_2$, $M \cap I_1 \neq \varnothing$, and we have $\tilde{v}_{ia} = \tilde{v}_{ib} = 0$ for every district $i \in M$. Hence,

$$s_a = 4y\ell - 4 \sum_{i \in M \cap I_1} x_i, \qquad s_b = 4y\ell + 4 \sum_{i \in M \cap I_1} x_i,$$

where x_i is the integer in X that corresponds to district D_i.

Now, if $\sum_{i \in M \cap I_1} x_i \neq 0$, either $s_a > 4y\ell + 4$ or $s_b > 4y\ell + 4$, in which case p cannot be the winner at $\tilde{\mathbf{v}}$. Thus, $\sum_{i \in M \cap I_1} x_i = 0$, and hence $X' = \{x_i \in X : i \in M \cap I_1\}$ is a witness that X is a yes-instance of Subset Sum.

Part (ii). To prove that PV-Man is NP-hard when the input is given in unary, we provide a reduction from X3C; see Definition 2.2.

Given an instance $\langle E, \mathcal{S} \rangle$ of X3C with $|E| = 3\ell$, $|\mathcal{S}| = s$, we construct an instance of PV-Man as follows. We set $C = \{j_e : e \in E\} \cup \{p\}$, where p is the attacker's preferred candidate. The districts are partitioned into three sets I_1, I_2, I_3:

- For each subset $S \in \mathcal{S}$ the set I_1 contains a district D_S. In this district each candidate j_e such that $e \in S$ gets 3ℓ votes, and all other candidates get no votes.
- For each element $e \in E$, the set I_2 contains $3\ell s + 9\ell^2 - 3\ell \cdot |\{S \in \mathcal{S} : e \in S\}|$ districts; each of these districts consists of a single voter who votes for j_e.

- The set I_3 contains a single district D^* that consists of $3\ell s - 2\ell$ voters who vote for p.

We set $B_{\mathcal{A}} = \ell$, $B_{\mathcal{D}} = 0$ and $\gamma_i = n_i$ for all $i \in [k]$.

We have $\mathrm{SW}(j_e) = 3\ell s + 9\ell^2$ for all $e \in E$ and $\mathrm{SW}(p) = 3\ell s - 2\ell$. Hence, the true winner is the candidate in $C \setminus \{p\}$ who is favored by the tie-breaking rule. We show that the attacker is able to make p the winner if and only if E admits an exact cover by sets from $\mathcal{S}$.

Suppose that $\mathcal{Q} \subseteq \mathcal{S}$ is an exact cover for E; note that $|\mathcal{Q}| = \ell$. The attacker can manipulate the ℓ districts in I_1 that correspond to sets in $\mathcal{Q}$ by reassigning all the 9ℓ votes in each of them to p. In the resulting election, p gets $3\ell s + 9\ell^2 - 2\ell$ votes, while every other candidate j_e gets $3\ell s + 9\ell^2 - 3\ell$ votes, as every e is covered by exactly one set in $\mathcal{Q}$.

Conversely, suppose the attacker has a successful manipulation $(M, \tilde{\mathbf{v}})$ with $|M| \leq \ell$. For each $c \in C$, let s_c denote the number of votes that c receives in $\tilde{\mathbf{v}}$. As p can gain at most 9ℓ votes for each district in M, we have $s_p \leq 3\ell s + 9\ell^2 - 2\ell$. Let $\mathcal{Q} = \{S \in \mathcal{S} : D_S \text{ is manipulated}\}$; note that $|\mathcal{Q}| \leq \ell$. We claim that $\mathcal{Q}$ is a cover for E. Indeed, if for some $e \in E$ no district in $\{D_S : e \in S\}$ is manipulated, the manipulation lowers the score of j_e by at most ℓ, so $s_{j_e} \geq 3\ell s + 9\ell^2 - \ell > s_p$, a contradiction. □

In the hardness reductions in the proof of Theorem 4.3 the defender's budget is 0. This indicates that the attacker's problem remains NP-hard, irrespective of the defender's budget and recounting behavior. For instance, our hardness result holds when the defender is known to use a heuristic, such as a greedy algorithm, to compute her response (indeed, when the budget is 0, all recounting strategies are equivalent to each other).

5 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.

Theorem 5.1. PD-Rec *is* NP-*complete even when*

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

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

Proof. This problem is clearly in NP. We give separate hardness proofs for the case $m = 3$ (part (i)) and for the case where the input is given in unary (part (ii)); again, the problem becomes polynomial-time solvable if both of the constraints (i) and (ii) are satisfied (Theorem 5.2).

Part (i). We use the same reduction as in the proof of the first part of Theorem 4.1. An important feature of this reduction is that all voters in each district vote for the same candidate. Thus, if we set the weight of each district to be equal to the number of voters therein, the proof goes through without change.

Part (ii). We provide a reduction from Independent Set; see Definition 2.3. Given an instance $\langle G, \ell \rangle$ of Independent Set, where $G = (V, E)$, we construct an instance of PD-Rec as follows. Let $\nu = |V|$, $\mu = |E|$; we can assume without loss of generality that $\mu \geq 1$. We set $C = \{j_u : u \in V\} \cup \{j_e : e \in E\} \cup \{a, p\}$, where p is the attacker's preferred candidate; thus, $|C| = \nu + \mu + 2$. We create the following districts, where each district consists of a single voter, and this voter's vote can be changed by the manipulator [7]:

[7] The number of voters in each district can be chosen arbitrarily; all that matters is that the manipulator can change the winner arbitrarily in each district.

- For each edge $e = \{x, y\} \in E$, there are two districts $D_{e,x}$ and $D_{e,y}$ with weight 2 each. In each such district $D_{e,u}$ the winner before manipulation is j_e, and the winner after manipulation is j_u.

- For each node $u \in V$, there is a district D_u with weight 2μ; in this district the winner before manipulation is j_u, and the winner after manipulation is p.

- There is a set I of $2(\nu + \mu) + 1$ districts with weight $\frac{2}{2(\nu+\mu)+1}$ each[8]; in each such district the winner before manipulation is a, and the winner after manipulation is p.

- There is a district of weight $2(\nu - \ell)\mu + 3$ with winner a; this district is not manipulated.

- For each $e \in E$, there is a district of weight $2(\nu - \ell)\mu$ with winner j_e; this district is not manipulated.

- For each $u \in V$, there is a district of weight $2(\nu - \ell)\mu - 2\mu + 2$ with winner j_u; this district is not manipulated.

The budget of the defender is $B_{\mathcal{D}} = \nu + \mu$. The candidates' weights before and after manipulation are given in the following table:

	true weight	distorted weight
a	$2(\nu - \ell)\mu + 5$	$2(\nu - \ell)\mu + 3$
p	0	$2\nu\mu + 2$
$j_e, e \in E$	$2(\nu - \ell)\mu + 4$	$2(\nu - \ell)\mu$
$j_u, u \in V$	$2(\nu - \ell)\mu + 2$	$\leq 2(\nu - \ell)\mu + 2$

Hence, the true winner is candidate a and the winner after manipulation is p.

If $V' \subseteq V$ is an independent set of size ℓ in G, the defender can proceed as follows. For each $u \in V'$, she demands a recount in D_u and in each district $D_{e,u}$ such that e is incident to u. Since V' forms an independent set, this requires recounting at most $\nu + \mu$ districts. Moreover, after the recount the weight of p is $2(\nu - \ell)\mu + 2$, the weight of a is $2(\nu - \ell)\mu + 3$, the weight of each candidate j_u such that $u \in V'$ is $2(\nu - \ell)\mu + 2$, the weight of each candidate j_u such that $u \in V \setminus V'$ is at most $2(\nu - \ell)\mu + 2$, and the weight of each candidate j_e such that $e \in E$ is at most $2(\nu - \ell)\mu + 2$. Thus, this recounting strategy successfully restores a as the election winner.

Conversely, suppose that the defender has a recounting strategy R that results in making a the election winner. Since $|R| \leq B_{\mathcal{D}}$, at most $\nu + \mu$ districts in I can be recounted, so a's weight after the recount is at most $2(\nu - \ell) + 3 + \frac{2(\nu+\mu)}{2(\nu+\mu)+1} < 2(\nu - \ell) + 4$. Now, if R contains at most $\ell - 1$ districts in $\{D_u : u \in V\}$, then p's weight after the recount is at least $2(\nu - \ell + 1)\mu + 2 \geq 2(\nu - \ell) + 4$, a contradiction with a becoming the winner after the recount. Hence, R contains at least ℓ districts in $\{D_u : u \in V\}$; let V' be the subset of nodes corresponding to these districts. We claim that V' forms an independent set in G.

Indeed, consider a node $u \in V'$. If the defender does not recount some district $D_{e,u}$ such that u is incident to e then after the recount the weight of j_u is at least $2(\nu - \ell)\mu + 4$, a contradiction with a becoming the winner after the recount. Thus $D_{e,u}$ is necessarily recounted. Now, suppose that $e = \{x, y\} \in E$ for some $x, y \in V$. We have just argued that both $D_{e,x}$ and $D_{e,y}$ have to be recounted. But this means that the score of j_e is at least $2(\nu - \ell)\mu + 4$ after the recount, a contradiction again. Thus, V' is an independent set. □

Theorem 5.2. PD-Rec *can be solved in time* $O(k \cdot B_{\mathcal{D}} \cdot (W + 1)^m)$*, where* $W = \sum_{i \in [k]} w_i$ *is the total weight of all districts.*

[8]For convenience, we use fractional weights. We can turn all weight into integers, by multiplying them by $2(\nu + \mu) + 1$.

Proof. The algorithm is a simple adaptation of the dynamic program presented in the proof of Theorem 4.2. In more detail, our algorithm now fills out a table T containing entries of the form $T(i, \ell, \mathbf{u})$, for each $i \in \{0, 1, \dots, k\}$, $\ell \in \{0, 1, \dots, B_\mathcal{D}\}$, and $\mathbf{u} = (u_a)_{a \in C} \in \{0, 1, \dots, W\}^m$; thus, $|T| = (k+1) \cdot (B_\mathcal{D} + 1) \cdot (W+1)^m$. We define $T(i, \ell, \mathbf{u}) = $ *true* if we can recount at most ℓ of the first i districts so that for each $a \in C$ it holds that the total weight of the districts candidate a wins equals u_a; otherwise we define $T(i, \ell, \mathbf{u}) = $ *false*. Once the table is filled out, we need to scan it to determine if there exists a recounting strategy that makes c the eventual winner, similarly to the proof of Theorem 4.2. □

We also obtain an easiness 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 5.3. 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 under the Plurality rule [Faliszewski, 2008], which is polynomial-time solvable; see Section 2 for the definition of this problem. 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 $\pi_{iy} = 0$, $\pi_{iz} = +\infty$ 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 $\pi_{ix} = 1$. 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. It turns out that for the PD rule we can obtain a stronger hardness result than for PV: we will now argue that when weights and vote counts are given in binary, PD-Man is Σ_2^P-complete even for $m = 3$.[9] Our reduction uses a variant of the Subset Sum problem, which we term Sub-Subset Sum (SSS); this problem may be of independent interest.

Definition 5.4 (Sub-Subset Sum)**.** An instance of Sub-Subset Sum 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 $X'' \subseteq X'$, and a no-instance otherwise.

Our proof proceeds by establishing that SSS is Σ_2^P-complete (Lemma 5.5; the proof can be found in the appendix), and then reducing this problem to PD-Man.

Lemma 5.5. *SSS is* Σ_2^P*-complete.*

Theorem 5.6. PD-Man *is* Σ_2^P*-complete, even when* $m = 3$.

Proof. 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 $(0, 0, 2y + 4 + \sum_{x \in X^-} 3x)$.

[9] Problems that are complete for Σ_2^P are widely believed to be harder than NP-hard problems; in particular, it is not known how to reduce them to the Satisfiability problem, so one cannot use SAT solvers to find solutions to them. The most prominent Σ_2^P-complete problem is to decide whether a Boolean formula of the form $(\exists x_1, \dots, x_n)(\forall y_1, \dots, y_n)\phi(x_1, \dots, x_n, y_1, \dots, y_n)$ is satisfiable. For formal definitions and further discussion, see, e.g., the textbook by Papadimitriou [1994].

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 non-empty subset $X'' \subseteq X'$ 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 $\mathrm{SW}(p) > \mathrm{SW}(b)$, 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 $3y + 7 + \sum_{x \in X^-} 3x \geq 2y + 7$. 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. □

We conjecture that PD-Man remains Σ_2^P-complete when the input is given in unary; however, for this setting we are only able to prove that this problem is NP-hard.

Theorem 5.7. PD-Man *is* NP*-hard, even when* $B_{\mathcal{D}} = 0$ *and the input vote profile and district weights are given in unary.*

Proof. To show that PD-Man is NP-hard even when the input votes and district weights are given in unary, we provide a reduction from Independent Set; see Definition 2.3.

Given an instance $\langle G, \ell \rangle$ of Independent Set with $G = (V, E)$, we construct the following instance of PD-Man. Let $\nu = |V|$, $\mu = |E|$. We set $C = \{j_u : u \in V\} \cup \{j_e : e \in E\} \cup \{a, p\}$, where p is the attacker's preferred candidate; thus, $|C| = \nu + \mu + 2$. Then, we create the following districts; the weight of each district is equal to the number of voters therein.

- For every edge $e = \{x, y\} \in E$, we create two districts $D_{e,x}$ and $D_{e,y}$ with 5 voters each; thus, $w_{e,x} = w_{e,y} = 5$. In each such district $D_{e,u}$ there are two voters who vote for j_e and three voters who vote for j_u. We set $\gamma_{e,u} = 1$; thus, the attacker can change the winner in this district from j_u to j_e.
- For every node $u \in V$, we create a district D_u with 5μ voters; thus, $w_u = 5\mu$. In each such district there are 2μ voters who vote for j_u and 3μ voters who vote for a. We set $\gamma_u = \mu$; thus, the attacker can change the winner in this district from a to j_u.
- There are also some districts that cannot be manipulated (i.e., $\gamma = 0$). We specify the weights and the winners of these districts.
  - For each $e \in E$, there is a district with weight $5\mu(\nu - \ell) - 5$ and winner j_e.
  - For each $u \in V$, there is a district with weight $5\mu(\nu - \ell - 1)$ and winner j_u.
  - Finally, there is a district with weight $5\mu(\nu - \ell) + 1$ and winner p.

The budgets are $B_{\mathcal{A}} = \nu + \mu$ and $B_{\mathcal{D}} = 0$.

We have $\mathrm{SW}(a) = 5\mu\nu$, $\mathrm{SW}(p) = 5\mu(\nu - \ell) + 1$, $\mathrm{SW}(j_e) = 5\mu(\nu - \ell) - 5$ for each $e \in E$, and $\mathrm{SW}(j_u) = 5\mu(\nu - \ell - 1) + 5\,|\{e \in E : u \in e\}| \leq 5\mu(\nu - \ell)$ for each $u \in V$. Hence, the true winner of the election is candidate a. We show that the attacker can make p the winner if and only if $\langle G, \ell \rangle$ is a yes-instance of Independent Set, i.e., there is an independent set of size ℓ in G.

Suppose first that there is an independent set $V' \subseteq V$, $|V'| = \ell$, in G. The following manipulation strategy makes p the winner. For every $u \in V'$, change the winner of district D_u from a to j_u, and for every $e \in E$ such that $u \in e$, change the winner of district $D_{e,u}$ from j_u to j_e. Note that since V' is an independent set, the weight of each candidate j_e, $e \in E$, increases by at most 5. Let ω_c denote the weight of each candidate $c \in C$ after manipulation. We have $\omega_a = 5\mu(\nu - \ell)$, $\omega_p = 5\mu(\nu - \ell) + 1$, $\omega_{j_e} \in \{5\mu(\nu - \ell) - 5, 5\mu(\nu - \ell)\}$ for each $e \in E$, and $\omega_{j_u} = 5\mu(\nu - \ell)$ for each $u \in V$; thus, candidate p becomes the winner of the election.

Conversely, suppose that the attacker has a manipulation that makes p the election winner; for each $c \in C$, let ω_c be the weight of candidate c after this manipulation. Since p cannot be made the winner in any additional district, we have $\omega_p = 5\mu(\nu - \ell) + 1$. Let V' be the set of all nodes $u \in V$ such that the attacker changes the winner of D_u from a to j_u. Since $\omega_a \leq \omega_p$, we have $|V'| \geq \ell$; we will now argue that V' is an independent set. Indeed, consider a node $u \in V'$. Changing the winner in D_u from a to j_u increases the weight of j_u by 5μ. As we have $\omega_{j_u} \leq \omega_p$, the manipulation needs to reduce the weight of j_u by $5|\{e \in E : u \in e\}|$. The only way to do so is to change the winner from j_u to j_e in all districts $D_{e,u}$ with $u \in e$, thereby increasing the weight of j_e by 5. Now, suppose that $x, y \in V'$ and $e = \{x, y\} \in E$. Then the manipulation increases the weight of j_e by 10, so we have $\omega_{j_e} = 5\mu(\nu - \ell) + 5 > \omega_p$, a contradiction. Thus, V' is an independent set. □

PD-Man remains NP-hard even if all districts have the same weight; however, under this assumption this problem can be placed in NP, i.e., the unweighted variant of PD-Man is strictly easier than its weighted variant unless NP$= \Sigma_2^P$ (which is believed to be highly unlikely).

Theorem 5.8. PD-Man *is* NP*-complete when* $w_i = 1$ *for all* $i \in [k]$.

Proof. To see that PD-Man is in NP when $w_i = 1$ for all $i \in [k]$, it suffices to note that PD-Rec is in P under this assumption (Theorem 5.3). To prove that PD-Man remains NP-hard even in this case, we again provide a reduction from Independent Set; see Definition 2.3.

Given an instance $\langle G, \ell \rangle$ of Independent Set, where $G = (V, E)$, we construct an instance of PD-Man as follows. Let $\nu = |V|$, $\mu = |E|$, and for each $u \in V$ let $\deg(u)$ denote the degree of vertex u in G; without loss of generality, we can assume that $\mu > 0$ and $\deg(u) > 0$ for all $u \in V$. Let $A_V = \{a_u : u \in V\}$, $A'_V = \{b_u : u \in V\}$, $A_E = \{a_e : e \in E\}$, and set $C = A_V \cup A'_V \cup A_E \cup \{p\}$, where p is the attacker's preferred candidate; thus, $|C| = 2\nu + \mu + 1$. The tie-breaking order $\succ$ is defined so that $p \succ c$ for all $c \in C \setminus \{p\}$, and $c \succ c'$ for all $c \in A_V$, $c' \in A_E$. We create the following districts (note that the weight of each district is 1).

- For every edge $e = \{x, y\} \in E$, we create two districts $D_{e,x}$ and $D_{e,y}$ with 5 voters each. In each such district $D_{e,u}$ there are two voters who vote for a_u and three voters who vote for a_e. We set $\gamma_{e,u} = 1$; thus, the attacker can change the winner in this district from a_e to a_u.

- For every vertex $u \in V$, we create a district D_u with two voters who vote for a_u and three voters who vote for b_u. We set $\gamma_u = 1$; thus, the attacker can change the winner in this district from b_u to a_u.

- There are also some districts that cannot be manipulated (i.e., $\gamma = 0$); for concreteness, we assume that each such district has five voters, and they all vote for the same candidate:

  - For each $e \in E$, there are $\mu - 1$ districts where the winner is a_e;

  - For each $u \in V$, there are $\mu - \deg(u)$ districts where the winner is a_u;
  - There are μ districts where the winner is p.

The budgets are $B_{\mathcal{A}} = 2\mu + \ell$ and $B_{\mathcal{D}} = \ell$. Thus, we have $\mathrm{SW}(p) = \mu$, $\mathrm{SW}(a_e) = \mu + 1$ for each $e \in E$, $\mathrm{SW}(a_u) = \mu - \deg(u) < \mu$ and $\mathrm{SW}(b_u) = 1$ for each $u \in V$. Consequently, the true winner is one of the candidates in A_E.

We will now argue that G admits an independent set of size ℓ if and only if there is a winning strategy for the attacker.

Suppose first that $V' \subseteq V$ is an independent set of size ℓ. Consider the following strategy for the attacker, which changes votes in exactly $B_{\mathcal{A}}$ districts:

- For each $e = \{x, y\} \in E$, change the winner of $D_{e,x}$ from a_e to a_x, and the winner of $D_{e,y}$ from a_e to a_y.
- For each $u \in V'$, change the winner of D_u from b_u to a_u.

Let ω_c denote the weight of each candidate $c \in C$ after this manipulation. We have $\omega_p = \mu$, $\omega_{a_e} = \mu - 1$ for each $e \in E$, $\omega_{a_u} = \mu$ for each $u \in V \setminus V'$, $\omega_{a_u} = \mu + 1$ for each $u \in V'$, and $\omega_{b_u} = 0$ each $u \in V$. Hence, in the manipulated instance the winner is chosen from $\{a_u : u \in V'\}$ according to the tie-breaking rule.

Even though p does not win the election at this point, we will now show that p becomes the winner once the defender respond optimally to this manipulation.

First, we show that the defender can make p win. To this end, for each $u \in V'$ the defender can pick one edge e^u such that $u \in e^u$ and demand a recount in district $D_{e^u,u}$; altogether, this strategy requires recounting $\ell = B_{\mathcal{D}}$ districts. Since V' is an independent set, after the recount the weight of each candidate a_e, $e \in E$, is at most μ, and also the weight of each candidate a_u, $u \in V$, is at most μ. Since $\omega_p = \mu$ and p is favored by the tie-breaking rule, p becomes the election winner.

We will now argue that for every candidate a that can be made the election winner by recounting at most ℓ districts we have $\mathrm{SW}(a) \leq \mathrm{SW}(p)$; since defender breaks ties according to $\succ$, this proves that the defender will choose a recounting strategy that makes p win. To see this, suppose for the sake of contradiction that there is a recounting strategy that results in a candidate a with $\mathrm{SW}(a) > \mathrm{SW}(p)$ becoming the election winner. Note that $\mathrm{SW}(a) > \mathrm{SW}(p)$ implies that $a \in A_E$ and hence $\omega_a = \mu - 1$. Let ω'_c denote the weight of each candidate $c \in C$ after the recount. The attacker does not transfer any district to a, which implies that $\omega'_a \leq \mathrm{SW}(a) = \mu + 1$. On the other hand, since $\omega'_p = \mu$, and the tie-breaking rule favors p over all other candidates, we have $\omega'_a \geq \mu + 1$. Thus, $\omega'_a = \mu + 1$. This means that $\omega'_a - \omega_a = 2$, i.e., if $a = a_e$ and $e = \{x, y\}$, both $D_{e,x}$ and $D_{e,y}$ are recounted. We will now argue that $x, y \in V'$. Indeed, for each $u \in V'$ we have $\omega_{a_u} = \mu + 1$; on the other hand, $a_u \succ a$ and hence $\omega'_{a_u} < \omega'_a = \mu + 1$. Thus, the defender must demand that for each $u \in V'$ the district D_u is recounted; since $B_{\mathcal{D}} = \ell$, the set of recounted districts is exactly V', and hence $x, y \in V'$, as claimed. But this is a contradiction, since $\{x, y\} \in E$, and V' is an independent set. This proves that if $\langle G, \ell \rangle$ is a yes-instance of Independent Set, there is a winning strategy for the attacker.

Conversely, suppose that G has no independent set of size ℓ. Consider an attack that changes votes in at most $B_{\mathcal{A}}$ districts. For each $c \in C$, let ω_c denote the weight of candidate c after the attack. Note that $\omega_p = \mu$; moreover, any attack can only increase the weight of candidates in A_V, and the weight of any such candidate after the attack is at most $\mu + 1$. Let $V' = \{u \in V : \omega_{a_u} = \mu + 1\}$ and $C' = \{a_u \in C : u \in V'\}$. We consider three cases:

- $|V'| > B_{\mathcal{D}}$. Since recounting a district only reduces the weight of one candidate, the weight of some candidate $a_u \in C'$ will still be $\mu + 1$ after the recount, so p will be beaten by a_u.
- $|V'| \leq B_{\mathcal{D}}$, V' is not an independent set. Pick an edge $e^* = \{x, y\}$ such that $x, y \in V'$, and consider the following recounting strategy. For each $u \in V' \setminus \{x, y\}$, the defender picks one

edge e^u such that $u \in e^u$, and demands a recount in districts $D_{e^u,u}$ for each $u \in V' \setminus \{x, y\}$ as well as in $D_{e^*,x}$ and in $D_{e^*,y}$. This recounting strategy requires recounting $|V'| \le B_{\mathcal{D}}$ districts, reduces the weight of every candidate $c \in C'$ by 1 and increases the weight of a_{e^*} by 2. Thus, after the recount the weight of a_{e^*} is $\mu + 1$, whereas the weights of all candidates in $C \setminus A_E$ do not exceed μ, so the winner is a candidate $a \in A_E$. Since $\text{SW}(a) > \text{SW}(p)$, this means that p cannot win after the recount.

- $|V'| \le B_{\mathcal{D}}$, V' is an independent set. Then by our assumption $|V'| < \ell = B_{\mathcal{D}}$. Consider an edge $e^* = \{x, y\}$ with $x \in V'$, $y \notin V'$. For each $u \in V' \setminus \{x\}$, the defender can pick one edge e^u such that $u \in e^u$, and demand a recount in districts $D_{e^u,u}$ for each $u \in V' \setminus \{x\}$ as well as in $D_{e^*,x}$ and in $D_{e^*,y}$. This strategy requires recounting $|V'| + 1 \le B_{\mathcal{D}}$ districts and ensures that after the recount the weight of e^* is $\mu + 1$, whereas the weights of all candidates in $C \setminus A_E$ are at most μ, so the winner is a candidate $a \in A_E$. Since $\text{SW}(a) > \text{SW}(p)$, this means that p cannot win after the recount.

Hence, the attacker cannot win in any case. This completes the proof. □

Theorem 5.7 holds even for $B_{\mathcal{D}} = 0$, but for Theorems 5.6 and 5.8 this is not the case. Indeed, there is evidence that the possibility of recounting may have impact on the complexity of the attacker's problem. Specifically, 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. Furthermore, the unweighted version of the attacker's problem can be shown to be in P when $B_{\mathcal{D}} = 0$.

To establish this, we first state and prove a simple lemma, which will also be used in the proof of Theorem 6.10. Effectively, this lemma shows that Plurality nonuniform bribery with unit prices remains easy even when the input is represented succinctly; our proof is inspired by the proof for weighted bribery in the work of Faliszewski et al. [2009].

Lemma 5.9. *Given a vote profile* $\mathbf{v}$ *over a candidate set* C *and a list of integers* $(\gamma_i)_{i\in[k]}$*, for each* $i \in [k]$ *let* C_i *be the set of all candidates that can be made winners in* D_i *by changing at most* γ_i *votes. Then the membership in* C_i *can be decided in polynomial time.*

Proof. Fix an $i \in [k]$. To decide whether a candidate a is in C_i, we first calculate a's Plurality score in D_i after it receives γ_i extra votes, i.e., $v_{ia} + \gamma_i$. Then, for each candidate $c \in C \setminus \{a\}$ we set $z_c = \max\{0, v_{ic} - v_{ia} - \gamma_i\}$ if $a \succ c$ and $z_c = \max\{0, v_{ic} - v_{ia} - \gamma_i + 1\}$ if $c \succ a$; the quantity z_c is the 'excess' score of candidate c compared to the final score of a (with tie-breaking taken into account). We then check if $\sum_{c\in C\setminus\{a\}} z_c \le \gamma_i$. If yes, we can ensure that a wins by transferring z_c votes from c to a for each $c \in C \setminus \{a\}$, and then, if $\gamma_i > \sum_{c\in C\setminus\{a\}}$, transferring arbitrary $\gamma_i - \sum_{c\in C\setminus\{a\}} z_c$ additional votes from other candidates to a. Otherwise, we have $a \notin C_i$: indeed, after any transfer of at most γ_i votes the score of a would be at most $v_{ia} + \gamma_i$, whereas reducing the score of all other candidates so that none of them beats a after the transfer would cost more than γ_i. □

We are now ready to describe our algorithm for PD-Man in the unweighted setting when the defender's budget is 0.

Proposition 5.10. PD-Man *can be solved in polynomial time if* $w_i = 1$ *for all* $i \in [k]$ *and* $B_{\mathcal{D}} = 0$.

Proof. Given a vote profile $\mathbf{v}$ over a candidate set C, a list of integers $(\gamma_i)_{i\in[k]}$, a preferred candidate p, and a budget $B_{\mathcal{A}}$, we transform an instance of our problem to an instance of Plurality nonuniform bribery as follows (for the definition of nonuniform bribery, see Section 2). Our bribery instance has the same set of candidates C and the same preferred candidate p. We map each district D_i to a single voter i. If the true winner in D_i is some candidate $c \in C$, we set $\pi_{ic} = 0$. Then for each $a \in C \setminus \{c\}$ we set $\pi_{ia} = 1$ if $a \in C_i$ and $\pi_{ia} = +\infty$ otherwise; by Lemma 5.9, membership in C_i can be decided efficiently. We set the bribery budget to be $B_{\mathcal{A}}$. It is immediate that the resulting instance is a 'yes'-instance of Plurality nonuniform bribery if and only if we started with a 'yes'-instance of PD-Man. □

6 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 other candidates. However, manipulations that give additional votes to candidates other than p are counter-intuitive 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 6.1 (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 6.2. We show an example for PV; this 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	$D_3, \ldots, D_8$	$D_9, \ldots, D_{12}$
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 $i \in \{1, \ldots, 12\}$. 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, 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 6.2 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, so asking them to implement a non-regular manipulation may cause confusion. Thus, it is interesting to understand if focusing on regular manipulations affects the complexity of the problems we consider.

In what follows, we consider the complexity of $\mathcal{R}$-Rec and $\mathcal{R}$-Man for $\mathcal{R} \in \{\mathrm{PV}, \mathrm{PD}\}$ under the assumption that the attacker is limited to regular manipulations; we denote these versions of our problems by $\mathcal{R}$-Rec-Reg and $\mathcal{R}$-Man-Reg, respectively. We first consider the defender's problem ($\mathcal{R}$-Rec-Reg) and then the attacker's problem ($\mathcal{R}$-Man-Reg).

Throughout this section, it will be convenient to partition C as $C = C_B^{\mathcal{R}} \cup \{p\} \cup C_G^{\mathcal{R}}$, where

$$C_B^{\mathcal{R}} = \{b \in C \setminus \{p\} : \mathrm{SW}^{\mathcal{R}}(b) < \mathrm{SW}^{\mathcal{R}}(p) \text{ or } \mathrm{SW}^{\mathcal{R}}(b) = \mathrm{SW}^{\mathcal{R}}(p), p \succ b\}$$

is the set of 'bad' candidates and

$$C_G^{\mathcal{R}} = \{g \in C \setminus \{p\} : \mathrm{SW}^{\mathcal{R}}(g) > \mathrm{SW}^{\mathcal{R}}(p) \text{ or } \mathrm{SW}^{\mathcal{R}}(g) = \mathrm{SW}^{\mathcal{R}}(p), g \succ p\}$$

is the set of 'good' candidates; we suppress the superscript $\mathcal{R}$ when it is clear from the context.

Before we proceed, we state an observation that will be useful for our analysis.

Proposition 6.3. *Let $\mathcal{R} \in \{\mathrm{PV}, \mathrm{PD}\}$, and let $(M, \tilde{\mathbf{v}})$ be a winning regular manipulation. Then, for every recounting strategy $R \subseteq M$ it holds that p is the election winner after the recount.*

Proof. Since M is a winning manipulation, the winner after recounting is either p or some candidate in C_B; we will show that when M is regular, the latter case is, in fact, impossible. For each $c \in C$, let s_c denote the number of votes/vote weight of c after the recount. Since M is a regular manipulation, for each candidate $b \in C_B$ we have

$$s_b \leq \mathrm{SW}^{\mathcal{R}}(b) \leq \mathrm{SW}^{\mathcal{R}}(p) \leq s_p,$$

and if $b \succ p$, the second inequality is strict. Thus, p beats every candidate in C_B after recounting, so no such candidate can be the election winner. □

By setting $R = \varnothing$ in Proposition 6.3, we observe that p is the winner at $\tilde{\mathbf{v}}$, i.e., the situation described in Example 6.2, where p does not win after the manipulation, but the defender is forced to make p the election winner, cannot occur if the attacker is limited to regular manipulations.

6.1 The Defender's Problem

We will soon see (Theorems 6.6 and 6.7) that the defender's problem remains NP-hard even if the attacker is limited to regular manipulations; this holds both for Plurality over Voters and for Plurality over Districts. Indeed, similarly to our results for the general case, most of our hardness results for the regular case hold even if there are only three candidates or if the input is given in unary; an important exception is Plurality over Districts with unary weights, for which the defender's problem admits a polynomial-time algorithm in the regular setting (Theorem 6.8).

However, before presenting these hardness results, we will now describe a natural greedy heuristic for a defender who is faced with a regular manipulation. While this heuristic may fail to identify an optimal recounting strategy, it can be used to decide whether the defender can make a 'good' candidate win (Lemma 6.4); also, it serves as a $1/2$-approximation algorithm for the defender's optimization problem (Theorem 6.5), and it will play an important role in our analysis of the attacker's problem (Section 6.2). Intuitively, for each 'good' candidate a our heuristic tries to recount the $B_{\mathcal{D}}$ districts that offer the maximum advantage to a; e.g., for PV the 'value' of district i for candidate a given a regular manipulation $(M, \tilde{\mathbf{v}})$ is measured as $(v_{ia} - v_{ip}) - (\tilde{v}_{ia} - \tilde{v}_{ip})$. We will now describe this heuristic in detail.

Let $\mathcal{R} \in \{\mathrm{PV}, \mathrm{PD}\}$, and consider a regular manipulation $(M, \tilde{\mathbf{v}})$. Recall that we assume that $|M| > B_{\mathcal{D}}$. By Proposition 6.3, we can assume that p is the winner at $\tilde{\mathbf{v}}$. Initially, the defender defines the set of *provisional winners* to consist of p. Then, for every candidate $a \in C_G$ the algorithm performs the following calculation. It sorts the districts in M in non-increasing order in terms 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 $g \in C_G$ (note that it may happen that $g \neq a$) winning the election, the defender adds g 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 6.4. *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.*

Proof. We provide the proof for $\mathcal{R} = \mathrm{PV}$; the proof for $\mathcal{R} = \mathrm{PD}$ is obtained by replacing candidates' vote counts with weights. Given a set of districts $R \subseteq M$ and a candidate $a \in C$, let $s_a(R)$ denote the vote count of candidate a after the attacker uses the manipulation $(M, \tilde{\mathbf{v}})$ and the defender recounts the districts in R. We write $s_a(R) \rhd s_b(R)$ if $s_a(R) > s_b(R)$ or $s_a(R) = s_b(R)$ and $a \succ b$; likewise, $\mathrm{SW}(a) \rhd \mathrm{SW}(b)$ if $\mathrm{SW}(a) > \mathrm{SW}(b)$ or $\mathrm{SW}(a) = \mathrm{SW}(b)$ and $a \succ b$.

Since $(M, \tilde{\mathbf{v}})$ is a regular manipulation, for every $R \subseteq M$ we have

$$s_p(R) \geq \mathrm{SW}(p) \quad \text{and} \quad s_c(R) \leq \mathrm{SW}(c) \text{ for all } c \in C \setminus \{p\}.$$

Suppose that $(M, \tilde{\mathbf{v}})$ is not a winning strategy for the attacker. Then there exists a subset R^* of at most $B_{\mathcal{D}}$ districts such that the winner after R^* is recounted is a 'good' candidate, i.e., some $a \in C_G$. We can assume without loss of generality that $|R^*| = B_{\mathcal{D}}$. Indeed, suppose that $|R^*| < B_{\mathcal{D}}$ and a wins after R^* is recounted. Let Q be an arbitrary set of $B_{\mathcal{D}}$ districts such that $R^* \subset Q \subseteq M$, and suppose that once the votes in Q are recounted, the winner is another candidate c; we will argue that $c \in C_G$. To see this, note that since $(M, \tilde{\mathbf{v}})$ is a regular manipulation, we have

$$\mathrm{SW}(c) \geq s_c(Q) \rhd s_a(Q) \geq s_a(R^*) \rhd s_p(R^*) \geq \mathrm{SW}(p).$$

Thus, $\mathrm{SW}(c) \rhd \mathrm{SW}(p)$, which means $c \in C_G$.

We will now argue that the greedy recounting algorithm does not output p. Let R_a be the set of districts recounted by this algorithm when it considers candidate a (i.e., R_a contains the top $B_{\mathcal{D}}$ districts in terms of the quantity $(v_{ia} - v_{ip}) - (\tilde{v}_{ia} - \tilde{v}_{ip})$), and let b be the winner after districts in R_a are recounted. Consider the following possibilities.

- $b \neq p$. If $b \in C_G$, the algorithm adds b to the set of provisional winners and thus does not output p. Otherwise, we have $b \in C_B$. However, this implies that $s_p(R_a) \geq \mathrm{SW}(p) \rhd \mathrm{SW}(b) \geq s_b(R_a)$, and hence, $s_p(R_a) \rhd s_b(R_a)$, which contradicts the assumption that b is the winner after R_a is recounted.

- $b = p$. By our choice of R_a and the fact that $|R_a| = |R^*|$ we have

$$\sum_{i \in R_a} ((v_{ia} - v_{ip}) - (\tilde{v}_{ia} - \tilde{v}_{ip})) \geq \sum_{i \in R^*} ((v_{ia} - v_{ip}) - (\tilde{v}_{ia} - \tilde{v}_{ip})).$$

  It follows that

$$\begin{aligned} s_a(R_a) - s_p(R_a) &= \sum_{i \in [k]} (\tilde{v}_{ia} - \tilde{v}_{ip}) + \sum_{i \in R_a} ((v_{ia} - v_{ip}) - (\tilde{v}_{ia} - \tilde{v}_{ip})) \\ &\geq \sum_{i \in [k]} (\tilde{v}_{ia} - \tilde{v}_{ip}) + \sum_{i \in R^*} ((v_{ia} - v_{ip}) - (\tilde{v}_{ia} - \tilde{v}_{ip})) \\ &= s_a(R^*) - s_p(R^*) \\ &\geq 0. \end{aligned}$$

  Since b is the winner after R_a is recounted, we have $s_p(R_a) \rhd s_a(R_a)$. Now that $s_a(R_a) - s_p(R_a) \geq 0$, it must be that $s_p(R_a) = s_a(R_a)$ and $p \succ a$. Thus, the inequality above must be an equality; in particular, $s_p(R^*) = s_a(R^*)$. Since $p \succ a$, it follows immediately that $s_p(R^*) \rhd s_a(R^*)$, which contradicts the assumption that a is the winner after R^* is recounted.

Therefore, we should always have $b \neq p$, in which case the algorithm does not output p. □

Notably, greedy recounting does not constitute an algorithm for $\mathcal{R}$-Rec-Reg: it is unable to decide whether there is a recounting strategy that results in a specific candidate becoming the election winner. However, we will now show that it 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 6.5. *Greedy recounting is a $1/2$-approximation algorithm for the optimization versions of* PV-Rec-Reg *and* PD-Rec-Reg.

Proof. We focus on PV; to adapt the analysis for PD it suffices to modify the notation so as to take into account the weights of the candidates rather than their vote counts.

Consider an instance with a set of candidates C, $|C| = m$, and let p be the attacker's preferred candidate. Suppose that the attacker uses a regular manipulation $(M, \tilde{\mathbf{v}})$; by Proposition 6.3, we can assume that p is the winner in the manipulated instance. For each $c \in C$, let s_c denote the vote count of candidate c in the manipulated instance. If p is the winner before the manipulation or if no recounting strategy can change the outcome, then greedy recounting is trivially optimal. Hence, we can assume that the defender can make some 'good' candidate win; let c be the defender's most preferred candidate with this property, and let R be a recounting strategy that results in c becoming the winner after a recount. We consider the round in which greedy recounting examines candidate c; suppose that greedy recounting selects a subset of districts Z. Let $A = C \setminus \{c, p\}$ denote the set of the remaining $m - 2$ candidates.

We define the following pairwise disjoint sets of districts:

- $I_Z = Z \setminus R$;
- $I_O = R \setminus Z$;
- $I_{OZ} = R \cap Z$;
- $I_{\overline{OZ}} = M \setminus (R \cup Z)$.

Given a set of districts $I \subseteq M$ and a subset of candidates $J \subseteq C \setminus \{p\}$, let

$$\Delta(I, J) = \sum_{i \in I} \sum_{a \in J} (v_{ia} - \tilde{v}_{ia})$$

denote the total number of votes in I that are transferred by the attacker from candidates in J to p; if J or I is a singleton, we omit the curly braces and write $\Delta(I, a)$ or $\Delta(i, J)$, respectively. Since $(M, \tilde{\mathbf{v}})$ is a regular manipulation, we have

$$s_p = \mathrm{SW}(p) + \Delta(M, c) + \Delta(M, A), \tag{1}$$

$$s_c = \mathrm{SW}(c) - \Delta(M, c), \tag{2}$$

$$s_a = \mathrm{SW}(a) - \Delta(M, a) \text{ for each } a \in A. \tag{3}$$

Since recounting the districts in $R = I_O \cup I_{OZ}$ ensures that c becomes the winner, we obtain

$$s_c + \Delta(I_O \cup I_{OZ}, c) \geq s_p - \Delta(I_O \cup I_{OZ}, c) - \Delta(I_O \cup I_{OZ}, A); \tag{4}$$

if $p \succ c$, this inequality is strict.

Next, let us focus on the behavior of the greedy recounting. Let $g \in I_Z$ and $o \in I_O$. Since the greedy algorithm selects g, but not o, we have

$$(v_{gc} - v_{gp}) - (\tilde{v}_{gc} - \tilde{v}_{gp}) \geq (v_{oc} - v_{op}) - (\tilde{v}_{oc} - \tilde{v}_{op}).$$

Substituting $v_{ic} - \tilde{v}_{ic} = \Delta(i, c)$ and $\tilde{v}_{ip} - v_{ip} = \Delta(i, c) + \Delta(i, A)$ for $i \in \{g, o\}$, we then obtain

$$2\Delta(g, c) + \Delta(g, A) \geq 2\Delta(o, c) + \Delta(o, A).$$

Since $|Z| = B_{\mathcal{D}}$, $|R| \leq B_{\mathcal{D}}$, we have $|I_Z| \geq |I_O|$. Pick a subset of districts $I'_Z \subseteq I_Z$ with $|I'_Z| = |I_O|$. We can pair each $o \in I_O$ with a unique $g \in I'_Z$, and add all corresponding inequalities to get

$$2\Delta(I'_Z, c) + \Delta(I'_Z, A) \geq 2\Delta(I_O, c) + \Delta(I_O, A);$$

since $\Delta(i, b) \geq 0$ for all $b \in A \cup \{c\}$ and all $i \in I_Z \setminus I'_Z$, we get

$$2\Delta(I_Z, c) + \Delta(I_Z, A) \geq 2\Delta(I_O, c) + \Delta(I_O, A). \tag{5}$$

By adding inequalities (4) and (5), we obtain

$$\begin{aligned} &s_c + \Delta(I_O \cup I_{OZ}, c) + 2\Delta(I_Z, c) + \Delta(I_Z, A) \\ &\geq s_p - \Delta(I_O \cup I_{OZ}, c) - \Delta(I_O \cup I_{OZ}, A) + 2\Delta(I_O, c) + \Delta(I_O, A), \end{aligned}$$

or, rearranging,

$$s_c + \Delta(I_Z \cup I_{OZ}, c) \geq s_p - \Delta(I_Z \cup I_{OZ}, c) - \Delta(I_Z \cup I_{OZ}, A); \tag{6}$$

if $p \succ c$, then Inequality (4) is strict and hence Inequality (6) is strict as well. Inequality (6) means that after recounting the districts in $Z = I_Z \cup I_{OZ}$, c beats p, i.e., the winner is either c or another candidate $a \in A$. To conclude the proof, it suffices to show that if the winner in the recounted instance is some candidate $a \in A$ then $\text{SW}(a) \geq \frac{1}{2}\text{SW}(c)$.

By substituting expressions for s_c and s_p from (2) and (1), we can write Inequality (6) as

$$\text{SW}(c) - \Delta(I_O \cup I_{\overline{OZ}}, c) \geq \text{SW}(p) + \Delta(I_O \cup I_{\overline{OZ}}, c) + \Delta(I_O \cup I_{\overline{OZ}}, A).$$

Since $\text{SW}(p) \geq 0$ and $\Delta(I_O \cup I_{\overline{OZ}}, A) \geq 0$, it follows that

$$\Delta(I_O \cup I_{\overline{OZ}}, c) \leq \frac{1}{2}\text{SW}(c). \tag{7}$$

By our assumption, recounting the districts in $Z = I_Z \cup I_{OZ}$ results in a getting at least as many votes as c, so we obtain

$$s_a + \Delta(I_Z \cup I_{OZ}, a) \geq s_c + \Delta(I_Z \cup I_{OZ}, c).$$

By substituting expressions for s_a and s_c from (3) and (2), we can rewrite this inequality as

$$\text{SW}(a) - \Delta(I_O \cup I_{\overline{OZ}}, a) \geq \text{SW}(c) - \Delta(I_O \cup I_{\overline{OZ}}, c).$$

Finally, since $\Delta(I_O \cup I_{\overline{OZ}}, a) \geq 0$, from Inequality (7) we obtain

$$\text{SW}(a) \geq \frac{1}{2}\text{SW}(c),$$

as desired. □

In fact, the bound on the approximation ratio provided by Theorem 6.5 is essentially tight.

Theorem 6.6. *For any constant* ε *with* $0 < \varepsilon \leq \frac{1}{2}$, *neither* PV-Rec-Reg *nor* PD-Rec-Reg *admit a polynomial-time* $(\frac{1}{2} + \varepsilon)$*-approximation algorithm unless* P = NP, *even when* $m = 3$.

Proof. We focus on PV; the proof for PD follows by setting the weight of each district in the reduction below to be equal to the number of voters therein.

Fix a positive constant ε with $0 < \varepsilon \leq \frac{1}{2}$. We will show that if there is a $(\frac{1}{2} + \varepsilon)$-approximation algorithm for PV-Rec-Reg, it can be used to solve Partition; see Definition 2.5.

Given an instance X of Partition with $|X| = \ell$, we construct an instance of PV-Rec-Reg with a set of candidates $C = \{a, b, p\}$, where p is the attacker's preferred candidate, as follows. Let $y = \sum_{x \in X} x$, and $z = \lceil y/\varepsilon \rceil$; note that $z \geq y$. Without loss of generality, we assume that all integers in X are divisible by 4 and hence $y \geq 4$. In what follows, we describe each district D_i by a tuple (v_{ia}, v_{ib}, v_{ip}). The districts are partitioned into the following three sets I_1, I_2 and I_3:

- For each $x \in X$, there is a district in I_1 with votes $(0, 2x\ell, 0)$, which are distorted to $(0, 0, 2x\ell)$.
- I_2 consists of $2z\ell$ districts with votes $(1, 0, 0)$, which are distorted to $(0, 0, 1)$.
- I_3 consists of two districts with votes $(2z\ell + y\ell + 2\ell, 0, 0)$ and $(0, 2z\ell, 0)$, which are not distorted.

Finally, the budget of the defender is $B_{\mathcal{D}} = \ell - 1$.

Since votes are transferred to p only, the manipulation is regular. The vote counts of the candidates before and after the manipulation are as follows:

	True vote counts (SW)	Distorted vote counts
a	$4z\ell + y\ell + 2\ell$	$2z\ell + y\ell + 2\ell$
b	$2z\ell + 2y\ell$	$2z\ell$
p	0	$2z\ell + 2y\ell$

Thus, before the manipulation the winner is a, and the manipulation makes p the election winner. Since $z\varepsilon \geq y$, we obtain $4z\ell\left(\frac{1}{2} + \varepsilon\right) \geq 2z\ell + 4y\ell$ and hence

$$\frac{\text{SW}(c)}{\text{SW}(a)} \leq \frac{2z\ell + 2y\ell}{4z\ell + y\ell + 2\ell} < \frac{1}{2} + \varepsilon \quad \text{for each } c \in \{b, p\}.$$

Therefore, any $(\frac{1}{2} + \varepsilon)$-approximation algorithm for PV-Rec-Reg can decide whether a can be restored as the winner. We will now argue that this is equivalent to deciding whether the given instance of Partition is a yes-instance.

Suppose that X is a yes-instance of Partition, i.e., there exists a subset $X' \subsetneq X$ such that $\sum_{x \in X'} x = y/2$; note that $|X'| \leq \ell - 1$. Then, by recounting the $|X'|$ districts of I_1 that correspond to the integers in X', the defender lowers the vote count of p by $y\ell$ and increases the vote count of b by $y\ell$. As a result, a gets $2z\ell + y\ell + 2\ell$ votes, b gets $2z\ell + y\ell$ votes, and p gets $2z\ell + y\ell$ votes. Therefore, a is restored as the election winner.

Conversely, suppose that there is no subset $X' \subseteq X$ such that $\sum_{x \in X'} x = y/2$. Since all integers in X are divisible by 4, $y/2$ is even and hence for any $X' \subseteq X$ we have $|\sum_{x \in X'} x - y/2| \geq 2$. Suppose that the defender recounts districts in I_1 that correspond to a subset $X' \subseteq X$ as well as q districts in I_2; let $u = \sum_{x \in X'} x$. Since $q < \ell$, the vote count of candidate a after the recount is

$$2z\ell + y\ell + 2\ell + q < 2z\ell + y\ell + 3\ell.$$

If $u \geq y/2 + 2$, then the vote count of b after the recount is

$$2z\ell + 2u\ell \geq 2z\ell + y\ell + 4\ell.$$

Otherwise, $u \leq y/2 - 2$ and hence $2\ell(y - u) \geq y\ell + 4\ell$. Since $q < \ell$, the vote count of p after the recount is

$$2z\ell - q + 2\ell \sum_{x \in X \setminus X'} x = 2z\ell - q + 2\ell(y - u)$$

$$> 2z\ell - \ell + y\ell + 4\ell$$
$$\geq 2z\ell + y\ell + 3\ell.$$

Therefore, in either case one of b or p gets more votes than a, and the theorem follows. □

Observe that the proof of Theorem 6.6 uses a reduction from PARTITION, a problem that becomes tractable when the input is represented in unary. We will now show that PV-REC-REG remains inapproximable within a factor strictly larger than $1/2$ even when the input is given in unary. Interestingly, for PD-REC-REG this is not the case; we will subsequently show (Theorem 6.8) that this problem becomes easy if the input given in unary.

Theorem 6.7. *For any constant* $\varepsilon > 0$ *with* $0 < \varepsilon \leq \frac{1}{2}$, PV-REC-REG *does not admit a polynomial-time* $(\frac{1}{2} + \varepsilon)$*-approximation algorithm unless* P = NP*, even when the input vote profile is given in unary.*

Proof. Fix an ε with $0 < \varepsilon \leq \frac{1}{2}$. We will show that a $(\frac{1}{2} + \varepsilon)$-approximation algorithm for PV-REC-REG can be used to solve the X3C problem; see Definition 2.2.

Given an instance $\langle E, \mathcal{S} \rangle$ of X3C with $|E| = 3\ell$, $|S| = s$, we construct an instance of PV-REC-REG as follows. We can assume without loss of generality that $s \geq \ell$. We set $C = \{j_e : e \in E\} \cup \{a, p\}$; thus, $|C| = 3\ell + 2$. Let $y = 10\lceil s/\varepsilon \rceil$. We create the following districts.

- There is a district D_S for each subset $S \in \mathcal{S}$. In this district each candidate $e \in S$ gets 4ℓ votes; the manipulation reallocates all these 12ℓ votes to p.
- There is a set of districts I, $|I| = y\ell - (6\ell s - 6\ell^2 + 2\ell)$; in each of this districts a gets 1 vote, and the manipulation reallocates this vote to p.
- There is a district with $y\ell + (6\ell s - 6\ell^2 + 2\ell)$ for a and $y\ell + (6\ell s - 6\ell^2 - 4\ell)$ votes for every $e \in E$, which is not manipulated.

The budget of the defender is set to $B_{\mathcal{D}} = \ell$. Since the votes are transferred to p only, this manipulation is regular. The vote counts of the candidates before and after the manipulation are as follows:

	True vote counts (SW)	Distorted vote counts
a	$2y\ell$	$y\ell + (6\ell s - 6\ell^2 + 2\ell)$
p	0	$y\ell + (6\ell s + 6\ell^2 - 2\ell)$
j_e	$\leq y\ell + (10\ell s - 6\ell^2 - 4\ell)$	$y\ell + (6\ell s - 6\ell^2 - 4\ell)$

Therefore, before the manipulation the winner is a, and the manipulation makes p the election winner. Since

$$\frac{\mathrm{SW}(c)}{\mathrm{SW}(a)} \leq \frac{y\ell + (10\ell s - 6\ell^2 - 4\ell)}{2y\ell} < \frac{y\ell + 10\ell s}{2y\ell} < \frac{1}{2} + \varepsilon$$

for all $c \in C \setminus \{a\}$, a $(\frac{1}{2} + \varepsilon)$-approximation algorithm for PV-REC-REG can be used to decide whether a can be restored as the election winner. We will now show that a can be restored as the winner if and only if our instance of X3C admits an exact cover.

To see this, suppose first that $\mathcal{Q} \subseteq \mathcal{S}$ is an exact cover of E; thus, $|\mathcal{Q}| = \ell$. The defender can recount the ℓ districts in $R = \{D_S : S \in \mathcal{Q}\}$. In the resulting election, a still gets $y\ell + (6\ell s - 6\ell^2 + 2\ell)$ votes, p gets $y\ell + (6\ell s - 6\ell^2 - 2\ell)$ votes, and the vote count of every candidate j_e, $e \in E$, increases by 4ℓ, so that after the recount j_e gets $y\ell + (6\ell s - 6\ell^2)$ votes. Thus, this recounting strategy restores a as the election winner.

Conversely, consider a recounting strategy R, $|R| \leq B_{\mathcal{D}} = \ell$, that makes a the election winner. For each candidate $c \in C$, let s_c denote the number of votes that c gets after the manipulation, and let s'_c denote the number of votes that c gets after the recount. We have $s_p - s_a = 12\ell^2 - 4\ell$. We claim that $|R \cap \{D_S : S \in \mathcal{S}\}| = \ell$: otherwise, the recount lowers the score of p by at most $12\ell(\ell-1)+1 \le 12\ell^2 - 11\ell$ and increases the score of a by at most ℓ, so $s'_p - s'_a \ge 6\ell$, a contradiction with a becoming a winner after the recount. Thus, $R \cap I = \varnothing$ and hence $s'_a = s_a = y\ell + (6\ell s - 6\ell^2 + 2\ell)$. Let $\mathcal{Q} = \{S \in \mathcal{S} : D_S \in R\}$. We have $|\mathcal{Q}| = \ell$, so it remains to argue that the sets in $\mathcal{Q}$ are pairwise disjoint. To see this, suppose that $S \cap S' \neq \varnothing$ for some $S, S' \in \mathcal{Q}$, and consider some candidate j_e such that $e \in S \cap S'$. We have $s'_{j_e} - s_{j_e} \ge 8\ell$, so $s'_{j_e} \ge y\ell + (6\ell s - 6\ell^2 + 4\ell) > s'_a$, a contradiction with a becoming the winner after the recount. This establishes that $\mathcal{Q}$ is an exact cover of E. □

For PD-Rec-Reg with unary weights we obtain an easiness result that, in contrast to Theorem 5.2, holds even for an unbounded number of candidates.

Theorem 6.8. PD-Rec-Reg *can be solved in time* $O(m \cdot k^2 \cdot W + m \cdot k^3)$*, where* $W = \sum_{i \in [k]} w_i$ *is the total weight of all districts.*

Proof. Let $(M, \tilde{\mathbf{v}})$ be the regular manipulation of the attacker. By Proposition 6.3, we can assume that the attacker's preferred candidate p is the winner in the manipulated election. We present an algorithm which, given a candidate $a \in C \setminus \{p\}$, decides whether a can be restored as the winner by recounting at most $B_\mathcal{D}$ districts.

For readability, we phrase our problem as a special case of the weighted version of the nonuniform bribery problem for the Plurality rule (see Section 2 for the definition of nonuniform bribery); we note that, to the best of our knowledge, the weighted version of this problem has not been considered in prior work. For each district D_i, $i \in [k]$, we create a voter i of weight w_i. Suppose that under truthful voting the winner in D_i is some candidate $c \in C$. If $i \notin M$ or $c = p$, we set $\pi_{ic} = 0$, $\pi_{ix} = +\infty$ for each $x \in C \setminus \{c\}$; this encodes the fact that district D_i cannot be affected by the recount. If $i \in M$ and $c \neq p$, we set $\pi_{ip} = 0$, $\pi_{ic} = 1$, $\pi_{ix} = +\infty$ for each $x \in C \setminus \{p, c\}$; this encodes the fact that a recount can change the winner in D_i from p to c. Then, our problem can be stated as follows: can a briber with budget $B_\mathcal{D}$ bribe some of the voters to make a the weighted Plurality winner, i.e., can the briber ensure that a has at least as much vote weight as any other candidate and strictly more vote weight than any candidate c such that $c \succ a$?

For each $c \in C \setminus \{p\}$, let V_c be the set of all voters $i \in [k]$ such that $\pi_{ic} = 1$, and for each $c \in C$ let V'_c be the set of all voters $i \in [k]$ such that $\pi_{ic} = 0$; let q_c denote the total weight of voters in V'_c.

As a first step, we will compute some quantities that will provide useful guidance to the briber later on. Specifically, for each $c \in C \setminus \{a, p\}$, each $h = 0, 1, \dots, |V_c|$, and each $U = 0, \dots, W$, let $t_c(h, U) = 1$ if there is a subset of V_c of size h whose vote weight is exactly U; otherwise, let $t_c(h, U) = 0$. These quantities can be computed by dynamic programming. Indeed, fix an arbitrary order of voters in V_c and denote the weight of the i-th voter in this order by ξ_i. For each $i = 0, 1, \dots, |V_c|$, each $h = 0, 1, \dots, |V_c|$ and each $U = 0, \dots, W$, let $\tau_c(i, h, U) = 1$ if there is a subset of the first i voters in V_c whose size is h and whose vote weight is exactly U, and let $\tau_c(i, h, U) = 0$ otherwise. We have $\tau_c(0, h, U) = 1$ if and only if $h = 0, U = 0$. Moreover, for $i \ge 1$ we have

$$\tau_c(i, h, U) = \begin{cases} \tau_c(i-1, h, U) & \text{if } U < \xi_i \\ \max\{\tau_c(i-1, h, U), \tau_c(i-1, h-1, U-\xi_i)\} & \text{otherwise.} \end{cases}$$

We can then set $t_c(h, U) = 1$ if and only if $\tau_c(|V_c|, h, U) = 1$.

For each $\ell = 0, \dots, \min\{|V_a|, B_\mathcal{D}\}$, we will check if the briber can achieve his goal by bribing exactly ℓ votes from V_a and at most $B_\mathcal{D} - \ell$ other voters; our instance of PD-Rec-Reg is a yes-instance if the answer to this question is 'yes' for at least one value of ℓ in this range.

Note that if the briber chooses to bribe ℓ voters from V_a, it is optimal for him to bribe the ℓ heaviest voters; such voters can be identified in time $O(k \log k)$. If the total weight of these voters is q, then the total weight of a after the bribery is $q_a + q$. Now, the briber needs to ensure that after the bribery the vote weight of every candidate $c \in C \setminus \{a\}$ is at most $q_a + q$; if $c \succ a$, this bound needs to be strengthened to $q_a + q - 1$. When bribing voters outside of V_a, the briber transfers vote weight from p to candidates in $C \setminus \{a, p\}$, so if $q_c > q_a + q$ or $q_c = q_a + q$, $c \succ a$, for some $c \in C \setminus \{a, p\}$, the briber cannot succeed; thus, from now we assume that this is not the case. For each $c \in C \setminus \{a, p\}$ let $z_c = q_a + q - q_c - 1$ if $c \succ a$ and $z_c = q_a + q - q_c$ if $a \succ c$. It follows that a becomes the winner after the bribery if and only if the bribery increases the vote weight of each candidate $c \in C \setminus \{a, p\}$ by at most z_c, and reduces the vote weight of p to at most $q_a + q$ (if $a \succ p$) or at most $q_a + q - 1$ (if $p \succ a$).

Given these observations, we are ready to solve our problem by dynamic programming. Renumber the candidates in $C \setminus \{a, p\}$ as $c_1, \ldots, c_{m-2}$. For each $j = 1, \ldots, m-2$, and each $h = 0, \ldots, |V_{c_j}|$, let $r_j(h) = \max\{U : t_{c_j}(h, U) = 1, U \le z_{c_j}\}$; the quantity $r_j(h)$ is the maximum amount of weight that can be safely transferred from p to c_j by bribing h voters in V_c. For each $j = 0, \ldots, m-2$, $i = 0, \ldots, B_{\mathcal{D}} - \ell$, let $T[i, j]$ be the maximum vote weight that can be moved from p to candidates in $\{c_1, \ldots, c_j\}$ by bribing at most i voters so that a still beats all candidates in $\{c_1, \ldots, c_j\}$ after the bribery. We have

$$T[i, j] = \begin{cases} 0, & \text{if } j = 0 \text{ or } i = 0 \\ \max\left\{T[i-h, j-1] + r_j(h) : 0 \le h \le i\right\}, & \text{otherwise} \end{cases}$$

Now, $T[B_{\mathcal{D}} - \ell, m-2]$ is the maximum vote weight that can be moved from p to candidates in $C \setminus \{a, p\}$ by bribing at most $B_{\mathcal{D}} - \ell$ voters so that a still beats all candidates in $C \setminus \{a, p\}$ after the bribery. Thus, it remains to check whether it holds that $q_p - T[B_{\mathcal{D}} - \ell, m-2] \le q_a + q$; if $p \succ a$, we additionally require that this inequality is strict. If this is indeed the case then a successful bribery—and hence a successful recounting strategy—exists.

The quantities $\tau_c(i, h, U)$ and $t_c(h, U)$ can be computed in time $O(k^2 \cdot W)$ for each $c \in C \setminus \{a, p\}$. Then, for each value of ℓ we can compute the quantities $r_j(h)$ in time $O(m \cdot k \cdot W)$. For each ℓ, each cell of the table $T[i, j]$ can be filled in time $O(k)$, and there are $O(m \cdot k)$ cells. As we perform this computation for each value of ℓ between 0 and $\min\{|V_a|, B_{\mathcal{D}}\}$, and we assume that $B_{\mathcal{D}} \le k$, the running time of our algorithm can be bounded as $O(m \cdot k^2 \cdot W + m \cdot k^3)$. □

An immediate corollary of Theorem 6.8 is that PD-Rec-Reg is in P when all districts have the same weight; of course, this also follows from the fact that PD-Rec is in P in the unweighted setting (Theorem 5.3).

6.2 The Attacker's Problem

Greedy recounting also plays an important role in our analysis of $\mathcal{R}$-Man-Reg. Indeed, even though greedy recounting does not constitute an algorithm for $\mathcal{R}$-Rec-Reg, Lemma 6.4 suffices to establish that $\mathcal{R}$-Man-Reg is in NP: the attacker can guess a regular manipulation and use greedy recounting to verify whether it is successful. For PV, this complexity upper bound is tight: one can check that in the hardness proofs in Theorem 4.3 the attacker's successful manipulation strategy is regular, and hence PV-Man-Reg is NP-complete. We summarize these observations in the following theorem.

Theorem 6.9. PV-Man-Reg *is* NP-*complete. The hardness result holds even if* $m = 3$ *or if the input vote profile is given in unary.*

In contrast, the hardness proofs for PD-Man (Theorems 5.6–5.8) rely on the attacker using a non-regular strategy, and therefore they do not show that PD-Man-Reg is NP-hard: in fact, it turns out that PD-Man-Reg is polynomial-time solvable, i.e., for PD focusing on regular manipulations brings down the complexity of the attacker's problem from Σ_2^P all the way down to P.

Theorem 6.10. PD-Man-Reg *can be solved in polynomial time.*

Proof. Let p be the attacker's preferred candidate. For each $c \in C \setminus \{p\}$, we denote by S_c the set of districts that have c as their true winner and can be manipulated in favor of p; by Lemma 5.9, this set can be computed efficiently.

Let $S = \bigcup_{c \in C \setminus \{p\}} S_c$ denote the set of all districts that can be manipulated in favor of p. Since the manipulation is regular, the attacker's strategy can be identified with a subset $M \subseteq S$. Let $\beta = \min\{B_{\mathcal{A}}, |S|\}$ be the maximum number of districts that can be manipulated.

Intuitively, our algorithm builds the set M incrementally in at most β steps, adding one district to M at each step. Let Q be the set of districts selected by the algorithm after q steps, $q < \beta$. At step $q+1$, the algorithm first tries a greedy approach: it adds to Q a set of $\beta - q$ heaviest districts in $S \setminus Q$ and checks (using greedy recounting, see Lemma 6.4) whether this manipulation is successful. If yes, the algorithm terminates; if not, greedy recounting identifies a 'good' candidate $a \in C_G$ such that after this manipulation the defender can recount so that a becomes the winner. In the latter case, instead of adding to Q the $\beta - q$ heaviest districts in $S \setminus Q$, the attacker targets candidate a, picking one of the heaviest districts in $S_a \setminus Q$ and adding it to Q. It then proceeds to the next step.

We will now describe our algorithm more formally and argue that it identifies a winning manipulation whenever it exists. For any set $Q \subseteq S$, $|Q| \leq \beta$, let $f(Q)$ be the set that consists of the $\beta - |Q|$ heaviest districts in $S \setminus Q$, with ties broken arbitrarily; thus, $|f(Q) \cup Q| = \beta$. Our analysis is based on the following lemma.

Lemma 6.11. *Consider a subset $Q \subset S$ such that there exists a winning regular manipulation M, $|M| \leq \beta$, with $Q \subset M$. Suppose that when the attacker manipulates the districts in $Q \cup f(Q)$, there is a candidate $a \in C_G$ such that the defender can make a beat p by recounting at most $B_{\mathcal{D}}$ districts. Let $S_a^{\max} = \arg\max_{j \in S_a \setminus Q} w_j$. Then*

(i) $S_a \setminus Q \neq \varnothing$, and

(ii) for each $i \in S_a^{\max}$ there is a winning regular manipulation M', $|M'| \leq \beta$, with $Q \cup \{i\} \subseteq M'$.

Before we prove this lemma, we will describe our algorithm in more detail and explain why its correctness follows from the lemma. The algorithm proceeds as follows.

1. Set $Q \leftarrow \varnothing$.

2. Apply greedy recounting to $Q \cup f(Q)$ to check whether $Q \cup f(Q)$ is a winning regular manipulation. If yes, terminate and return $Q \cup f(Q)$. Otherwise greedy recounting returns a candidate $a \in C_G$ such that the defender can make a beat p by recounting at most $B_{\mathcal{D}}$ districts.

3. If $S_a \setminus Q = \varnothing$ or $|Q| = \beta$, then output $\varnothing$. Otherwise, select an arbitrary $i \in S_a^{\max}$, set $Q \leftarrow Q \cup \{i\}$, and go back to Step 2.

By Lemma 6.4, if the algorithm returns $Q \cup f(Q)$ at the end of Step 2, then $Q \cup f(Q)$ is a winning regular manipulation. Otherwise, by Lemma 6.11, there is no winning strategy. This shows that our algorithm is correct. To see that it runs in polynomial time, note that every execution of Step 2 increases $|Q|$ by 1, and $|Q|$ is bounded from above by $\beta \leq B_{\mathcal{A}}$.

To complete the proof, it remains to prove Lemma 6.11

Proof of Lemma 6.11. Suppose that there exist Q, M and a that satisfy the conditions in the statement of the lemma. For each candidate $c \in C$ and each $X \subseteq S$, let $s_c(X)$ denote the weight of c after the districts in X have been manipulated in favor of p. We prove each claim of the lemma separately.

Proof of claim (i). We will prove a stronger claim, namely, that $M \cap (S_a \setminus Q) \neq \varnothing$.

Suppose for the sake of contradiction that $M \cap (S_a \setminus Q) = \varnothing$. We will argue that in this case if the attacker manipulates $Q \cup f(Q)$ districts and the defender recounts at most $B_{\mathcal{D}}$ of these districts, in the resulting instance p always beats a, which contradicts the assumptions of the lemma that the defender can make a beat p. To prove this, we pick an arbitrary recounting strategy $R' \subseteq Q \cup f(Q)$, $|R'| \leq B_{\mathcal{D}}$, match it with a recounting strategy $R \subseteq M$ against manipulation M, and show that R' cannot make a beat p as long as R cannot. Indeed, R cannot make a beat p as M is a winning manipulation by assumption.

Fix a recounting strategy $R' \subseteq Q \cup f(Q)$ with $|R'| \leq B_{\mathcal{D}}$. We construct a recounting strategy R by taking the part $R' \cap Q$ and complementing it with a set R'' that contains the heaviest districts in $M \setminus Q$ (with ties broken arbitrarily), i.e., $R = (R' \cap Q) \cup R''$, so we have $R \subseteq M$ given that $Q \subset M$. To keep the size of R within the budget $|B_{\mathcal{D}}|$, we let R'' consist of only the $\min\{|R' \cap f(Q)|, |M \setminus Q|\}$ heaviest districts in $M \setminus Q$ (which ensures that $|R| \leq |R' \cap Q| + |R' \cap f(Q)| \leq |R'| \leq B_{\mathcal{D}}$). We will now show that

(a) $s_a((Q \cup f(Q)) \setminus R') \leq s_a(M \setminus R)$;

(b) $s_p((Q \cup f(Q)) \setminus R') \geq s_p(M \setminus R)$.

Once (a) and (b) hold, subtracting them gives

$$s_a((Q \cup f(Q)) \setminus R') - s_p((Q \cup f(Q)) \setminus R') \leq s_a(M \setminus R) - s_p(M \setminus R),$$

so we have $s_a(M \setminus R) \leq s_p(M \setminus R)$ given that M is a winning manipulation (and if $a \succ p$, this inequality is strict). As desired, the defender cannot make a beat p by recounting the districts in R' after the attacker manipulates districts in $Q \cup f(Q)$.

To prove (a), note that since $Q \subset M$, the assumption $M \cap (S_a \setminus Q) = \varnothing$ implies $M \cap S_a \subseteq Q$ and $(M \setminus Q) \cap S_a = \varnothing$. Since $M \subseteq S$, we have

$$\sum_{i \in M \setminus Q} w_{ia} = \sum_{i \in (M \setminus Q) \cap S_a} w_i = 0.$$

Moreover, since $Q \setminus R = (Q \setminus (R' \cap Q)) \cap (Q \setminus R'') = Q \setminus R'$, we have

$$\begin{aligned} s_a\big((Q \cup f(Q)) \setminus R'\big) &= \mathrm{SW}(a) - \sum_{i \in (Q \cup f(Q)) \setminus R'} w_{ia} \\ &\leq \mathrm{SW}(a) - \sum_{i \in Q \setminus R'} w_{ia} \\ &= \mathrm{SW}(a) - \sum_{i \in Q \setminus R} w_{ia} - \sum_{i \in M \setminus Q} w_{ia} \\ &\leq \mathrm{SW}(a) - \sum_{i \in Q \setminus R} w_{ia} - \sum_{i \in (M \setminus Q) \setminus R} w_{ia} \\ &= s_a(M \setminus R). \end{aligned}$$

Thus, (a) holds.

We will now prove (b). First, we claim that

$$\sum_{i \in f(Q) \setminus R'} w_i \geq \sum_{i \in (M \setminus Q) \setminus R} w_i. \tag{8}$$

Indeed, recall that $|R''| = \min\{|M \setminus Q|, |R' \cap f(Q)|\}$. Now, if $|R''| = |M \setminus Q|$, then $M \setminus Q \subseteq R$, so the right-hand side of this inequality is 0, and our claim is immediate. Otherwise, we have $|R''| = |R' \cap f(Q)|$. Recall that R'' is the part of R taken from $M \setminus Q$, so we have

$$R'' = R \cap (M \setminus Q),$$

which also contains the heaviest $|R''|$ districts in $M \setminus Q$. On the other hand, $|f(Q)| = \beta - |Q|$ and $|M| \leq \beta$, so

$$|M \setminus Q| \leq \beta - |Q| = |f(Q)|.$$

Now since $f(Q)$ contains the heaviest $\beta - |Q|$ districts in $S \setminus Q$, we can think of the sum on the left of (8) as taking the weights of the heaviest $|f(Q)|$ districts in $S \setminus Q$ and then removing $|R' \cap f(Q)|$ of these weights. On the other hand, the sum on the right is obtained by taking the weights of some $|M \setminus Q| \leq |f(Q)|$ districts in $M \setminus Q \subseteq S \setminus Q$ and then removing from them $R \cap (M \setminus Q) = R''$ which contains the largest $|R''| = |R' \cap f(Q)|$ of these weights. Hence, inequality (8) follows.

Now, we can write

$$\begin{aligned} s_p\big((Q \cup f(Q)) \setminus R'\big) &= \text{SW}(p) + \sum_{i \in Q \setminus R'} w_i + \sum_{i \in f(Q) \setminus R'} w_i \\ &\geq \text{SW}(p) + \sum_{i \in Q \setminus R} w_i + \sum_{i \in (M \setminus Q) \setminus R} w_i \\ &= s_p(M \setminus R). \end{aligned}$$

This establishes (b).

Proof of claim (ii). We have established that $S_a \setminus Q \neq \varnothing$ and hence $S_a^{\max} \neq \varnothing$. Now, suppose for the sake of contradiction that for some $i \in S_a^{\max}$ there is no winning regular manipulation M' with $|M'| \leq \beta$, $Q \cup \{i\} \subseteq M'$. Since all districts in $S_a^{\max}$ are identical from both the attacker's and the defender's perspective, it holds that, in fact, for every $i \in S_a^{\max}$ there is no winning regular manipulation M' with $|M'| \leq \beta$, $Q \cup \{i\} \subseteq M'$.

In the proof of claim (i), we have argued that $M \cap (S_a \setminus Q) \neq \varnothing$; pick some $j \in M \cap (S_a \setminus Q)$. Since $Q \cup \{j\} \subseteq M$ and M is a winning regular manipulation, it follows that $j \notin S_a^{\max}$. Pick some $i \in S_a^{\max}$ and set $M' = (M \setminus \{j\}) \cup \{i\}$. We will now obtain a contradiction by showing that M' is a winning regular manipulation. Consider an arbitrary recounting strategy $R' \subseteq M'$, $|R'| \leq B_{\mathcal{D}}$.

(a) If $i \in R'$, let $R = (R' \setminus \{i\}) \cup \{j\}$ so that $|R| = |R'| \leq B_{\mathcal{D}}$, and $M' \setminus R' = M \setminus R$. Since M is a winning strategy, for every $c \in C_G$ we have

$$s_c(M' \setminus R') - s_p(M' \setminus R') = s_c(M \setminus R) - s_p(M \setminus R) \leq 0;$$

if $c \succ p$, this inequality is strict.

(b) If $i \notin R'$, let $R = R'$. Then for every $c \in C \setminus \{a, p\}$ we have

$$s_c(M' \setminus R') = s_c(M \setminus R).$$

In addition, since both i and j are in $S_a \setminus Q$ but $j \notin S_a^{\max}$, we have $w_j < w_i$. Hence,

$$\begin{aligned} s_a(M' \setminus R') &= s_a(M \setminus R) + w_j - w_i < s_a(M \setminus R), \\ s_p(M' \setminus R') &= s_p(M \setminus R) - w_j + w_i > s_p(M \setminus R). \end{aligned}$$

Combining these facts, for every $c \in C_G$ we have

$$s_c(M' \setminus R') - s_p(M' \setminus R') < s_c(M \setminus R) - s_p(M \setminus R) \leq 0.$$

Thus, both in case (a) and in case (b), p remains the winner after recounting.

This completes the proof of the lemma. □

We have described an algorithm that finds a winning regular manipulation (and returns ∅ if no such manipulation exists) in polynomial time. Thus, PD-Man-Reg is in P. □

To conclude this section, we identify a scenario where the attacker can safely limit himself to regular manipulations only. Specifically, we show that if all districts have the same weight, the attacker can change all votes in each manipulated district, and his budget is not too large, then under the PD rule an attacker has a successful manipulation if and only if he has a successful regular manipulation; it follows that in this case PD-Man admits a polynomial-time algorithm.

Proposition 6.12. *Consider an instance of* PD-Man *where (i)* $w_i = 1$ *for each* $i \in [k]$*, (ii)* $\gamma_i = n_i$ *for each* $i \in [k]$*, and (iii) the preferred candidate* p *wins at most* $k - B_{\mathcal{A}}$ *districts. This instance is a yes-instance of* PD-Man *if and only if it is a yes-instance of* PD-Man-Reg.

Proof. Clearly, a yes-instance of PD-Man-Reg is a yes-instance of PD-Man. Conversely, suppose that the attacker has a successful strategy $(M, \tilde{\mathbf{v}})$ with $|M| \le B_{\mathcal{A}}$ that is not regular; we can assume without loss of generality that the manipulator changes the winner of each district in M. We will now convert $(M, \tilde{\mathbf{v}})$ into a regular manipulation $(M', \tilde{\mathbf{u}})$ as follows. For each $i \in M$, let a_i be the winner in D_i before the manipulation, and let $M_0 = \{i \in [k] : a_i \neq p\}$. By our assumption on the attacker's budget we have $|M_0| \ge B_{\mathcal{A}}$. Let M' be an arbitrary subset of $|M|$ districts in M_0 that contains all districts in $M \cap M_0$. We define $\tilde{\mathbf{u}}$ so that for each $i \in M'$ the manipulator changes all votes in D_i in favor of p.

Intuitively, to obtain the regular manipulation $(M', \tilde{\mathbf{u}})$ from the original manipulation $(M, \tilde{\mathbf{v}})$, we first change the votes in each manipulated district in favor of p. For some districts (namely, those where the original manipulation changed the winner from p to another candidate) this means that we 'undo' the manipulation in this district, thereby freeing up one unit of the budget; we spend this budget to change additional districts in favor of p. Our assumption on the attacker's budget ensures that this last step is feasible, i.e., there are sufficiently many additional districts that can be changed in favor of p.

By construction, the strategy $(M', \tilde{\mathbf{u}})$ is regular; we will now argue that it is successful. To this end, suppose for the sake of contradiction that there exists a recounting strategy $R' \subseteq M'$, $|R'| \le B_{\mathcal{D}}$, such that after the attacker manipulates the districts in M' in favor of p and the defender recounts the districts in R', the winner is some 'good' candidate $a \in C_G$. Let R be a subset of M of size $|R'|$ such that $M \cap R' \subseteq R$; note that $|M| = |M'|$, $|R| = |R'|$ and hence $|M \setminus R| = |M' \setminus R'|$. We will argue that if the attacker manipulates according to $(M, \tilde{\mathbf{v}})$ and the defender recounts the districts in R, candidate a beats p as well as all candidates in C_B. This implies that the winner after the recounting is a candidate in C_G and hence $(M, \tilde{\mathbf{v}})$ is not a winning strategy, a contradiction.

For each $c \in C$, let s'_c denote the weight of c when the attacker changes all districts in M' in favor of p and the defender recounts the districts in R', and let s_c denote the weight of c when the attacker manipulates according to $(M, \tilde{\mathbf{v}})$ and the defender recounts the districts in R.

Since a wins when the attacker manipulates the districts in M' in favor of p and the defender recounts the districts in R', we have $s'_a \ge s'_p$; if $p \succ a$, this inequality is strict. On the other hand, we have $s_p \le s'_p$ and $s_a \ge s'_a$. Combining these inequalities, we obtain

$$s_p \le s'_p \le s'_a \le s_a;$$

if $p \succ a$, the second inequality is strict and hence $s_p < s_a$. Thus, if the attacker manipulates according to $(M, \tilde{\mathbf{v}})$ and the defender recounts the districts in R, it holds that a beats p.

Now, consider a 'bad' candidate $b \in C_B$. We have $s_b \le \text{SW}(b) + |M \setminus R| = \text{SW}(b) + |M' \setminus R'| \le \text{SW}(p) + |M' \setminus R'| = s'_p$; as $b \in C_B$, if $b \succ p$ then the second inequality is strict. As a consequence, we obtain

$$s_b \le s'_p \le s'_a \le s_a;$$

moreover, if $s_b = s_a$, it follows that $p \succ b$ and $a \succ p$ and hence by transitivity $a \succ b$. Thus, if the attacker manipulates according to $(M, \tilde{\mathbf{v}})$ and the defender recounts the districts in R, it holds that a beats b. This completes the proof. □

Combining Proposition 6.12 and Theorem 6.10, we conclude that PD-Man is in P whenever all districts have the same weight, the attacker can change all votes in each manipulated district, and the attacker's budget is at most $k - \sum_{i \in [k]} w_{ip}$.

Corollary 6.13. PD-Man *can be solved in polynomial time if (i)* $w_i = 1$ *for each* $i \in [k]$*, (ii)* $\gamma_i = n_i$ *for each* $i \in [k]$*, and (iii) in the input instance, the preferred candidate* p *wins at most* $k - B_{\mathcal{A}}$ *districts.*

We will now establish that each of the conditions (i), (ii) and (iii) in Proposition 6.12 is necessary. Indeed, Example 6.2 shows this for condition (i): the reader can verify that the instance in this example satisfies conditions (ii) and (iii), but violates (i). Our next two examples demonstrate that conditions (ii) and (iii) are necessary as well.

Example 6.14. Consider an instance with 3 candidates $\{a, b, p\}$ and 35 voters who are distributed to 7 districts of unit weight. The vote profile is as follows:

Candidate	$D_1, \dots, D_4$	$D_5, \dots, D_7$
a	3	0
b	2	0
p	0	5

Also, $B_{\mathcal{A}} = 3$, $B_{\mathcal{D}} = 1$, and $\gamma_i = 1$ for all $i \in [7]$; note that we have $k - \sum_{i \in [k]} w_{ip} = 4 > B_{\mathcal{A}}$. The true winner is candidate a with weight 4, compared to the weight 3 and 0 of p and b, respectively. Since the attacker can only change one vote per district, there is no regular manipulation that can make p win. In contrast, the non-regular manipulation that distorts one vote in each of the first three districts so that b is the winner there (with 3 votes compared to the 2 votes of a) is a winning strategy for the attacker. Indeed, since the defender can only recount one district, her optimal strategy is to ensure that p is the winner by recounting one of the manipulated districts (if the tie-breaking rule favors p over b, the defender also has the option of not recounting any districts). □

Example 6.15. Consider an instance with 3 candidates $\{a, b, p\}$ and two single-voter districts, where a wins the first district, p wins the second district, and $w_1 = w_2 = 1$, $\gamma_1 = \gamma_2 = 1$. Suppose that the tie-breaking rule is $b \succ a \succ p$, so a wins by tie-breaking, and $B_{\mathcal{A}} = 2$, $B_{\mathcal{D}} = 1$. Note that for this instance we have $k - B_{\mathcal{A}} = 0$, so condition (iii) of Proposition 6.12 is not satisfied. The only regular manipulation available to the attacker is to change the winner in the first district to p, and the defender can thwart this manipulation by recounting that district. However, the attacker has a winning non-regular manipulation, where he changes the winner in the first district to p and the winner in the second district to b. Indeed, in the manipulated instance the winner is b by the tie-breaking rule. The defender can recount the first district, but that would not change the election winner; alternatively, she can recount the second district so that p becomes the winner. Since in the original election p wins one district and b wins zero districts, the defender prefers the second option, thereby making p the winner. □

Now, when it comes to polynomial-time solvability of PD-Man (Corollary 6.13), Theorem 5.8 illustrates the importance of condition (ii): the instance constructed in the hardness reduction in the proof of that theorem satisfies conditions (i) and (iii). We do not have a similar justification for condition (i), but Theorems 5.3 and 5.8 indicate that the setting with unit weights tends to be more tractable than the general setting. In contrast, the role of condition (iii) is less clear. While it is necessary for the reduction from PD-Man to PD-Man-Reg to go through, it might be possible to solve PD-Man with

$w_i = 1$, $\gamma_i = n_i$ for all $i \in [k]$ in polynomial time even if condition (iii) is violated, by developing a better understanding of the structure of non-regular manipulations in this setting. Indeed, while conditions (i) and (ii) capture important features of the input, condition (iii), while realistic, is nevertheless an artifact of our proof strategy; determining whether it is necessary for tractability is an interesting direction for future work.

7 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. A concrete open question is whether our Σ_2^P-hardness result holds if $\gamma_i = n_i$ for all $i \in [k]$. We note that a very recent paper by Gowda et al. [2020] investigates the parameterized complexity of the problems we study; the parameters considered in their work include the number of voters and the budgets of the attacker and the defender.

We contrasted our 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.

While we make the assumption of full information in our work, one may also consider probabilistic settings, or other forms of uncertainty. A simple extension is to assume uncertainty in the defender's prior knowledge; for example, the defender may only know an interval in which the true vote count of each district is contained, in which case the attacker can safely change the vote count within that interval without the defender realizing that the district has been manipulated. While our hardness results would still apply to any such generalized model, it would be interesting to see if our easiness results can be extended to more realistic settings.

References

P. Berman, M. Karpinski, L. L. Larmore, W. Plandowski, and W. Rytter. On the complexity of pattern matching for highly compressed two-dimensional texts. *Journal of Computer and System Sciences*, 65(2):332–350, 2002.

R. Bredereck, P. Faliszewski, R. Niedermeier, and N. Talmon. Large-scale election campaigns: Combinatorial shift bribery. *Journal of Artificial Intelligence Research*, 55:603–652, 2016.

L. Chen, L. Xu, S. Xu, Z. Gao, N. Shah, Y. Lu, and W. 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.

V. Conitzer and T. Walsh. Barriers to manipulation in voting. In *Handbook of Computational Social Choice*, pages 127–145. 2016.

P. Dey, N. Misra, S. Nath, and G. Shakya. A parameterized perspective on protecting elections. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 238–244, 2019.

E. Elkind, P. Faliszewski, and A. Slinko. Swap bribery. In *Proceedings of the 2nd International Symposium on Algorithmic Game Theory (SAGT)*, pages 299–310, 2009.

E. Elkind, J. Gan, S. Obraztsova, Z. Rabinovich, and A. A. Voudouris. Protecting elections by recounting ballots. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 259–265, 2019.

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

P. Faliszewski and J. Rothe. Control and bribery in voting. In *Handbook of Computational Social Choice*, pages 146–168. 2016.

P. Faliszewski, E. Hemaspaandra, and L. Hemaspaandra. How hard is bribery in elections? *Journal of Artificial Intelligence Research*, 35:485–532, 2009.

M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman, 1979.

K. N. Gowda, N. Misra, and V. Patel. A parameterized perspective on attacking and defending elections. In *Proceedings of the 31st International Workshop on Combinatorial Algorithms (IWOCA)*, volume 12126, pages 277–288, 2020.

J. A. Halderman and V. 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.

H. Kellerer, U. Pferschy, and D. Pisinger. *Knapsack problems.* Springer, 2004.

Y. Li, Y. Jiang, and W. Wu. Protecting elections with minimal resource consumption. In *Proceedings of the 16th Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, pages 1595–1597, 2017.

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

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

C. H. Papadimitriou. *Computational complexity.* Addison-Wesley, 1994.

C. 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.

D. Springall, T. Finkenauer, Z. Durumeric, J. Kitcat, H. Hursti, M. MacAlpine, and J. A. 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.

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

V. V. Vazirani. *Approximation algorithms.* Springer, 2001.

Y. Yin, Y. Vorobeychik, B. An, and N. Hazon. Optimal defense against election control by deleting voter groups. *Artificial Intelligence*, 259:32–51, 2018.

A Appendix: Proof of Lemma 5.5

In order to prove that SSS is Σ_2^P-complete, we will first show (Lemma A.1) that a variant of this problem, which we call SSS$^+$, is Σ_2^P-complete. We will then explain how to reduce SSS$^+$ to SSS.

An instance of SSS$^+$ is given by a multiset Y of *positive* integers and two integers q and s, $0 \leq q \leq |Y|$, $0 \leq s \leq \sum_{y \in Y} y$. It is a yes-instance if there exists a subset $Y' \subseteq Y$ with $|Y'| = q$ such that for all $Y'' \subseteq Y \setminus Y'$ it holds that $\sum_{y \in Y'} y + \sum_{y \in Y''} y \neq s$, and a no-instance otherwise.

Lemma A.1. SSS$^+$ *is* Σ_2^P*-complete.*

Proof. It is easy to see that SSS$^+$ is in Σ_2^P. In the remainder of the proof, we show that this problem is Σ_2^P-hard. We show a reduction from the BILEVEL SUBSET SUM (BI-SS) problem, which is known to be Σ_2^P-complete [Berman et al., 2002].

Definition A.2 (BILEVEL SUBSET SUM (BI-SS))**.** An instance of BI-SS is given by a positive integer t and two sets of positive integers A and B. It is a yes-instance if there exists a set $A' \subseteq A$ such that for all $B' \subseteq B$ it holds that $\sum_{a \in A'} a + \sum_{b \in B'} b \neq t$, and a no-instance otherwise.

It is convenient to think of both BI-SS and SSS$^+$ as leader-follower games. The leader acts first by selecting a subset; his aim is to prevent the sum of the integers chosen by both players from reaching a given target. The follower acts second; her aim is to select a subset so that the sum of the chosen integers equals the target. The difference between these two games is that in the former game the leader and the follower select from two different sets and there is no limit of the number of integers each of them can choose, whereas in the latter game the leader is limited to q integers and both parties choose from the same base set.

Given an instance $\langle A, B, t \rangle$ of BI-SS, where $A = \{a_1, \ldots, a_{|A|}\}$, $B = \{b_1, \ldots, b_{|B|}\}$, we proceed as follows. We will represent a positive integer x as a vector of bits of length $L = \lfloor \log_2 x \rfloor + 1$, denoted $\overline{x} = \overline{x^1 \ldots x^L}$: we have $\sum_{i \in [L]} \overline{x^i} \cdot 2^{L-i} = x$. We will consider numbers that correspond to bit vectors consisting of $2|B| + 2$ sections, with each section consisting of

$$\left\lceil \log_2 \left(\sum_{a \in A} a + (|A| + 1) \sum_{a \in B} b + 1 \right) \right\rceil + \lceil \log_2(2|A| + 2) \rceil$$

bits; this value, which is polynomial in the size of the input, is chosen so that addition operations do not carry bits across sections. For $h = 1, \ldots, 2|B| + 2$, let $\overline{x}(h)$ denote the h-th section of $\overline{x}$.

We now construct an instance of SSS$^+$ described by a triple $\langle Y, q, s \rangle$. Let $q = |A|$. The set Y consists of the following integers (see also Figure 1):

- For each $i = 1, \ldots, q$, there is an integer x_i such that $\overline{x_i}(1) = \overline{1}$, $\overline{x_i}(2|B|+2) = \overline{a_i}$, and $\overline{x_i}(h) = \overline{0}$ for each section $h \neq 1, 2|B| + 2$.

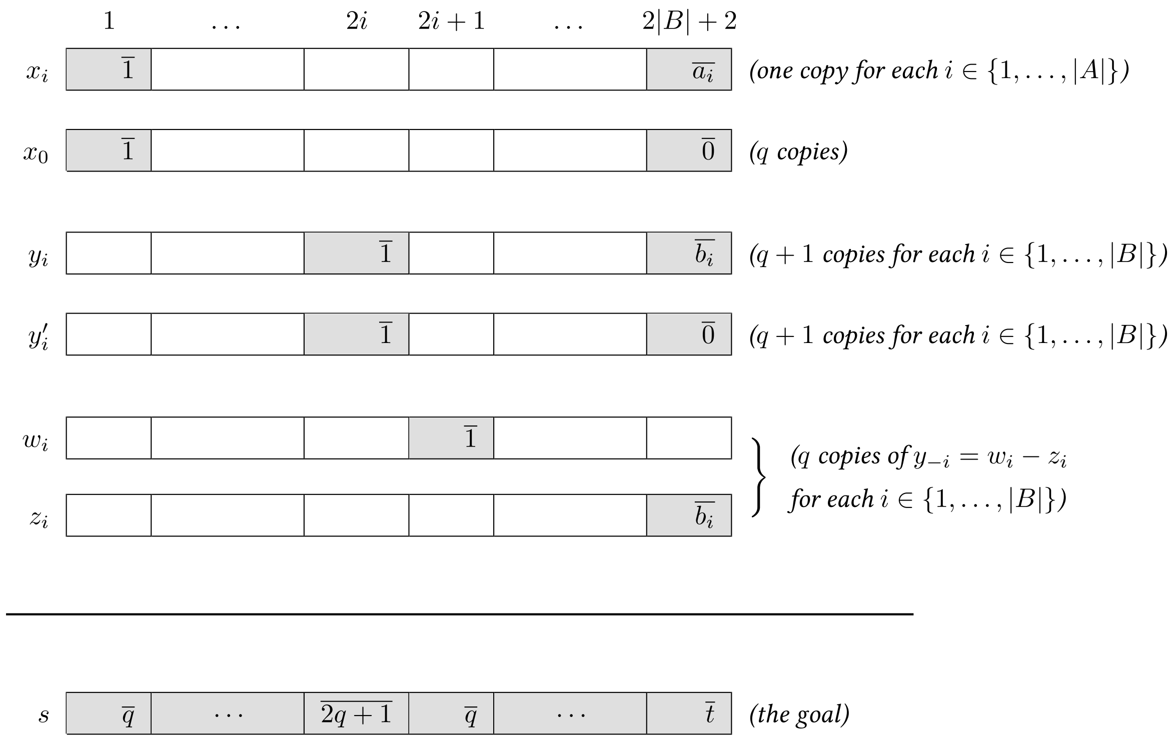

Figure 1: Reduction from BI-SS to SSS$^+$ (all blank sections are $\overline{0}$s).

- There are q copies of integer x_0 such that $\overline{x_0}(1) = \overline{1}$, and $\overline{x_0}(h) = \overline{0}$ for each section $h \neq 1$.
- For each $i = 1, \dots, |B|$, there are:
  - $q+1$ copies of integer y_i such that $\overline{y_i}(2i) = \overline{1}$, $\overline{y_i}(2|B|+2) = \overline{b_i}$, and $\overline{y_i}(h) = \overline{0}$ for each section $h \neq 2i, 2|B|+2$.
  - $q+1$ copies of integer y'_i such that $\overline{y'_i}(2i) = \overline{1}$ and $\overline{y'_i}(h) = \overline{0}$ for every section $h \neq 2i$.
  - q copies of integer $y_{-i} = w_i - z_i$, where w_i is such that $\overline{w_i}(2i+1) = \overline{1}$ and $\overline{w_i}(h) = \overline{0}$ for every $h \neq 2i+1$, while z_i is such that $\overline{z_i}(2|B|+2) = \overline{b_i}$ and $\overline{z_i}(h) = \overline{0}$ for every $h \neq 2|B|+2$.

Also, we set the goal s so that $\overline{s}(1) = \overline{q}$, $\overline{s}(2|B|+2) = \overline{t}$, and $\overline{s}(2h) = \overline{2q+1}$, $\overline{s}(2h+1) = \overline{q}$ for each $h \in \{1, \dots, |B|\}$.

To verify the correctness of the reduction, we first make the following observation. In the SSS$^+$ instance, the follower can achieve the goal s only if, for each $i = 1, \dots, |B|$, all copies of y_{-i} and exactly $2q+1$ out of the $2q+2$ copies of y_i and y'_i are included in the set $Y' \cup Y''$, which is chosen by the joint efforts of the leader and the follower: otherwise, the $2i$-th and the $(2i+1)$-st sections of the sum would not match the corresponding sections in s. The follower can decide whether $Y' \cup Y''$ will contain $q+1$ copy of y_i and q copies of y'_i or vice versa, since the leader's choice is restricted to q integers, while the follower's choice is unrestricted. Therefore, for each $i = 1, \dots, |B|$, the $(2|B|+2)$-nd section of the sum of the selected copies of y_i, y'_i and y_{-i} will be either $\overline{0}$ or $\overline{b_i}$; effectively, the follower chooses whether to include b_i in the sum.

Now, suppose that in the given BI-SS instance there exists a subset $A' \subseteq A$ such that for all $B' \subseteq B$ it holds that $\sum_{a \in A'} a + \sum_{b \in B'} b \neq t$. Then, in the corresponding instance of SSS$^+$ the leader can choose the subset Y' containing all x_i such that $a_i \in A'$ and $q - |A'|$ copies of x_0. Given this choice of the leader, the follower can only choose integers from the copies of y_i, y'_i, and y_{-i} since any other choice will make the first section of the sum different from $\overline{q}$. However, since $\sum_{a\in A'} a + \sum_{b\in B'} b \neq t$ for all $B' \subseteq B$, no matter which integers the follower chooses, the last section of the sum cannot be $\bar{t}$. Thus, this instance of SSS$^+$ is a yes-instance.

Conversely, suppose that the BI-SS INSTANCE is such that for every $A' \subseteq A$ there exists a $B' \subseteq B$ such that $\sum_{a\in A'} a + \sum_{b\in B'} b = t$. We will now argue that in the corresponding instance of SSS$^+$ the follower can always achieve the goal s. Indeed, suppose the leader chooses a set Y'. Let $A' = \{a_i : x_i \in Y'\}$, and, for each $i \in \{1, \ldots, |B|\}$, let α_i be the number of copies of y_i in Y', let α'_i be the number of copies of y'_i in Y', and let β_i be the number of copies of y_{-i} in Y'. Fix some set $B' \subseteq B$ such that $\sum_{a\in A'} a + \sum_{b\in B'} b = t$. To achieve the goal s, the follower can include the following integers in Y'':

- $q - |A'|$ copies of x_0, so that the first section of the sum is exactly $\overline{q}$;
- $q + 1 - \alpha_i$ copies of y_i, $q - \alpha'_i$ copies of y'_i, and $q - \beta_i$ copies of y_{-i} for each i such that $b_i \in B'$, so that the last sections of the copies of y_i, y'_i, and y_{-i} in $Y' \cup Y''$ sum up to $\overline{b_i}$;
- $q - \alpha_i$ copies of y_i, $q + 1 - \alpha'_i$ copies of y'_i, and $q - \beta_i$ copies of y_{-i} for each i such that $b_i \notin B'$, so that the last sections of the copies of y_i, y'_i, and y_{-i} in $Y' \cup Y''$ sum up to $\overline{0}$.

This completes the proof. $\square$

We are now ready to show that SSS is Σ_2^P-complete. This problem is obviously in Σ_2^P. We show that it is Σ_2^P-hard via a reduction from SSS$^+$. Given an instance $\langle Y, q, s\rangle$ of SSS$^+$, we construct an instance $\langle X, \ell\rangle$ of SSS as follows. Let $z = \sum_{y\in Y} y + 1$ and $z' = s - (q+2)z + 1$.

- Let X consist of all integers in Y, $2q+1$ copies of z, and $q+1$ copies of z'. Thus, $|X| = |Y|+3q+2$.
- Set $\ell = |Y| + 2q + 2$.

Observe that any subset $X' \subseteq X$ of size ℓ must contain: (1) at least $|Y| - q$ integers from Y, (2) at least one copy of z', and (3) at least $q + 1$ copies of z. We will show that $\langle Y, q, s\rangle$ is a yes-instance of SSS$^+$ if and only if $\langle X, \ell\rangle$ is a yes-instance of SSS.

Suppose that $\langle X, \ell\rangle$ is a yes-instance of SSS; thus, there exists $X' \subseteq X$, $|X'| = \ell$, such that $\sum_{x\in X''} x \neq 0$ for all non-empty subset $X'' \subseteq X'$. By our observation, $|X' \cap Y| \geq |Y| - q$. Consider a set Y' obtained by removing from Y exactly $|Y| - q$ elements in $X' \cap Y$; thus, $|Y'| = q$. We claim that $\sum_{y\in Y'} y + \sum_{y\in Y''} y \neq s$ for all $Y'' \subseteq Y \setminus Y'$. Indeed, pick an arbitrary subset $Y'' \subseteq Y \setminus Y'$, and consider a set X'' consisting of: all integers in $Y \setminus (Y'' \cup Y')$, one copy of z', and $q + 1$ copies of z. If $\sum_{y\in Y'} y + \sum_{y\in Y''} y = s$, then we have

$$\sum_{x\in X''} x = \sum_{x\in Y} x - \sum_{x\in Y''\cup Y'} x + z' + (q+1)z = z - 1 - s + z' + (q+1)z = 0,$$

which contradicts the assumption that $\langle X, \ell\rangle$ is a yes-instance of SSS. Therefore, $\sum_{y\in Y'} y + \sum_{y\in Y''} y \neq s$ for any arbitrary $Y'' \subseteq Y \setminus Y'$, and Y' witnesses that $\langle Y, q, s\rangle$ is a yes-instance of SSS$^+$.

Conversely, suppose that $\langle Y, q, s\rangle$ is a yes-instance of SSS$^+$, i.e., there exists a set $Y' \subseteq Y$ such that $|Y'| = q$ and $\sum_{y\in Y'} y + \sum_{y\in Y''} y \neq s$ for all $Y'' \subseteq Y \setminus Y'$. We show that $\langle X, \ell\rangle$ is a yes-instance of SSS, that is witnessed by the subset $X' \subseteq X$ consisting of all the $|Y| - q$ integers in $Y \setminus Y'$, all the $q + 1$ copies of z' and $2q + 1$ copies of z; obviously, $|X'| = \ell$.

Suppose for the sake of contradiction that $\sum_{x\in X''} x = 0$ for some non-empty subset $X'' \subseteq X'$. Since all integers in Y and z are positive, X'' must contain at least one copy of z' for the sum to be zero. In fact, X'' can contain at most one copy of z' because otherwise we will get the following contradiction:

$$\sum_{x\in X''} x \leq 2z' + (2q+1)\cdot z + \sum_{x\in Y} x = 2s - 2\sum_{x\in Y} x - 1 < 0.$$

Suppose X'' contains λ copies of z. We have

$$0 = \sum_{x \in X''} x = z' + \lambda \cdot z + \sum_{x \in X'' \cap (Y \setminus Y')} x = s + 1 + (\lambda - (q+2)) \cdot z + \sum_{x \in X'' \cap (Y \setminus Y')} x.$$

Rearranging the terms gives

$$(\lambda - (q+1)) \cdot z = z - s - 1 - \sum_{x \in X'' \cap (Y \setminus Y')} x.$$

We have $0 \leq s \leq \sum_{x \in Y} x$ as required in the definition of SSS$^+$, the right-hand side is in $(-z, z)$, so the only possibility is $\lambda - (q+1) = 0$ as all the numbers are integers. It follows that $z - s - 1 - \sum_{x \in X'' \cap (Y \setminus Y')} x = 0$, that is,

$$\begin{aligned} s = z - 1 - \sum_{x \in X'' \cap (Y \setminus Y')} x &= \sum_{x \in Y} x - \sum_{x \in X'' \cap (Y \setminus Y')} x \\ &= \sum_{x \in Y} x - \left(\sum_{x \in Y \setminus Y'} x - \sum_{x \in (Y \setminus Y') \setminus X''} x \right) \\ &= \sum_{x \in Y'} x + \sum_{x \in (Y \setminus Y') \setminus X''} x. \end{aligned}$$

Note that $(Y \setminus Y') \setminus X''$ is a subset of $Y \setminus Y'$, which implies that $\langle Y, q, s \rangle$ is a no-instance and contradicts our assumption. Therefore, $\sum_{x \in X''} x \neq 0$ for all non-empty subset $X'' \subseteq X'$, and X' witnesses that $\langle X, \ell \rangle$ is a yes-instance of SSS. $\square$