Computational Complexity Characterization of Protecting Elections from Bribery *

Lin Chen[1], Ahmed Sunny[1] Lei Xu[2], Shouhuai Xu[3], Zhimin Gao[4], Yang Lu[5], Weidong Shi[5], and Nolan Shah[6]

[1] Texas Tech University, 2500 Broadway, Lubbock, TX 79409, USA
lin.chen@ttu.edu ahmed.sunny@ttu.edu
[2] University of Texas Rio Grande Valley,1201 W University Dr, Edinburg, TX 78539, USA xuleimath@gmail.com
[3] University of Texas San Antonio,1 UTSA Circle, San Antonio, TX 78249, USA
shouhuai.xu@utsa.edu
[4] Auburn University at Montgomery,7430 East Dr, Montgomery, AL 36117, USA
mtion@masn.com
[5] University of Houston, 4800 Calhoun Rd, Houston, TX 77004, USA
ylu17@central.edu wshi3@uh.edu
[6] Amazon Web Services, Seatle, USA nolan@0x9b.com

Abstract. The bribery problem in election has received considerable attention in the literature, upon which various algorithmic and complexity results have been obtained. It is thus natural to ask whether we can protect an election from potential bribery. We assume that the protector can protect a voter with some cost (e.g., by isolating the voter from potential bribers). A protected voter cannot be bribed. Under this setting, we consider the following bi-level decision problem: Is it possible for the protector to protect a proper subset of voters such that no briber with a fixed budget on bribery can alter the election result? The goal of this paper is to give a full picture on the complexity of protection problems. We give an extensive study on the protection problem and provide algorithmic and complexity results. Comparing our results with that on the bribery problems, we observe that the protection problem is in general significantly harder. Indeed, it becomes Σ_2^p-complete even for very restricted special cases, while most bribery problems lie in NP. However, it is not necessarily the case that the protection problem is always harder. Some of the protection problems can still be solved in polynomial time, while some of them remain as hard as the bribery problem under the same setting.

Keywords: Voting, complexity, NP-hardness, Σ_2^p-hardness

*A 2 page extended abstract has been published at the Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems (AAMAS'18)

1 Introduction

In an election, there are a set of candidates and a set of voters. Each voter has a *preference list* of candidates. Given these preference lists, a winner is determined based on some voting rule, examples of which will be elaborated later.

In the context of election, the *bribery problem* has received considerable attention (see, for example, [1,7,8,12,14]). In this problem, there is an attacker who attempts to manipulate the election by bribing some voters, who will then report preference lists of the attacker's choice (rather than the voters' own preference lists). Each voter has a price for being bribed, and the attacker has an attack budget for bribing voters. There are two kinds of attackers: *constructive* vs. *destructive*. A *constructive* attacker attempts to make its designated candidate win an election, whereas the designated candidate would not win the election should there be no attacker. In contrast, a *destructive* attacker attempts to make its designated candidate lose the election, whereas the designated candidate would win the election should there be no attacker. The research question is: Given an attack budget for bribing, whether or not a (constructive or destructive) attacker can achieve its goal?

In this paper, we initiate the study of a new problem, called the *protection problem*, which extends the bribery problem as follows. There are also a set of candidates, a set of voters, and a bribery attacker. Each voter also has a *preference list* of candidates. There is also a voting rule according to which a winner is determined. Going beyond the bribery problem, the protection problem further considers a defender, who aims to protect elections from bribery. More specifically, the defender is given a defense budget and can use the defense budget to award some of the voters so that they cannot be bribed by the attacker anymore. This leads to an interesting problem: *Given a defense budget, is it possible to protect an election from an attacker with a given attack budget for bribing voters (i.e., assuring that the attacker cannot achieve its goal)?*

Our contributions. We introduce the problem of protecting elections from bribery, namely the protection problem. Given a defense budget for rewarding some of the voters and an attack budget for bribing some of the rest voters, the protection problem asks whether or not the defender can protect the election. We investigate the protection problem against the aforementioned two kinds of bribery attackers: constructive vs. destructive.

We present a characterization on the computational complexity of the protection problem (summarized in Table 1 in Section 5). The characterization is primarily concerning the voting rule of r-approval, which will be elaborated in Section 2. At a high level, our results can be summarized as follows. (i) The *protection problem* is hard and might be much harder than the *bribery problem*. For example, the protection problem is Σ_2^p-complete in most cases, while the bribery problem is in NP under the same settings. (ii) The *destructive protection problem* (i.e., protecting elections against a destructive attacker) is *no* harder than the *constructive protection problem* (i.e., protecting elections against a constructive attacker) in all of the settings we considered. In particular, the destructive

protection problem is Σ_2^p-hard only when the voters are weighted and have arbitrary prices, while the constructive protection problem is Σ_2^p-hard even when the voters are unweighted and have the unit price. (iii) Voter weights and prices have completely different effects on the computational complexity of the protection problem. For example, the constructive protection problem is coNP-hard in one case but is in P in another case.

Related Work. The problem of protecting elections from attacks seemingly has not received the due attention. Very recently, Yin et al. [20] considered the problem of defending elections against an attacker who can delete (groups of) voters. That is, the investigation is in the context of the *control problem*, where the attacker attempts to manipulate an election by adding or deleting some voters. The control problem has been extensively investigated (see, for example, [4,9,10,20]). Although the control problem is related to the bribery problem, the means used by the attacker in the control problem (i.e., attacker adding or deleting some voters) is different from the means used by the attacker in the bribery problem (i.e., attacker changing the preference lists of the bribed voters). We investigate the protection problem, which is defined in the context of the bribery problem rather than the control problem. That is, the problem we investigate is different from the problem investigated by Yin et al. [20].

The protection problem we study is inspired by the bribery problem. Faliszewski et al. [8] gave the first characterization on the complexity of the *bribery problem*, including some dichotomy theorems. In the bribery problem, the attacker can pay a fixed, but voter-dependent, price to arbitrarily manipulate the preference list of a bribed voter. The complexity of the bribery problem under the scoring rule of r-approval or r-veto for small values of r was addressed later by Lin [15] and Bredereck and Talmon [2]. There are also studies on measuring the bribery price in different ways (see, e.g., [1,6,12]).

Technically, the protection problem is related to the *bi-level optimization* problem, especially the *bi-level knapsack* problem ([3,5,17,18]). In the bi-level knapsack problem, there is a leader and a follower. The leader makes a decision first (e.g., packing a subset of items into the knapsack), and then the follower solves an optimization problem given the leader's decision (e.g., finding the most profitable subset of items that have not been packed by the leader). The problem asks for the decision of the leader such that a certain objective function is optimized (e.g., minimizing the profit of the follower). The protection problem we study can be formulated as the bi-level problem by letting the defender award some voters who therefore cannot be bribed by the attacker anymore, and then the attacker bribes some of the remaining voters as an attempt to manipulate the election.

2 Problem definition

Election model. Consider a set of m candidates $\mathcal{C} = \{c_1, c_2, \ldots, c_m\}$ and a set of n voters $\mathcal{V} = \{v_1, v_2, \ldots, v_n\}$. Each voter v_j has a preference list of candidates, which is essentially a permutation of candidates, denoted as τ_j. The

preference of v_j is denoted by $(c_{\tau_j(1)}, c_{\tau_j(2)}, \ldots, c_{\tau_j(m)})$, meaning that v_j prefers candidate $c_{\tau_j(z)}$ to $c_{\tau_j(z+1)}$, where $z = 1, 2, \ldots$. Since τ_j is a permutation over $\{1, 2, \ldots, m\}$, we denote by τ_j^{-1} the inverse of τ_j, meaning that $\tau_j^{-1}(i)$ is the position of candidate c_i in vector $(c_{\tau_j(1)}, c_{\tau_j(2)}, \ldots, c_{\tau_j(m)})$.

Voting rules. In this paper, we focus on the scoring rule (or scoring protocol) that maps a preference list to a m-vector $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_m)$, where $\alpha_i \in \mathbb{N}$ is the score assigned to the i-th candidate on the preference list of voter v_j and $\alpha_1 \geq \alpha_2 \geq \ldots \geq \alpha_m$. Given that τ_j is the preference list of v_j, candidate $c_{\tau_j(i)}$ receives a score of α_i from v_j. The *total score* of a candidate is the summation of the scores it received from the voters. The winner is the candidate that receives the highest total score. We focus on a *single-winner* election, meaning that only one winner is selected. In the case of a tie, a random candidate with the highest total score is selected. However, our results remain valid for all-natural variation of selecting a single winner.

We say a scoring rule is *non-trivial*, if $\alpha_1 > \alpha_m$ (i.e., not all scores are the same). There are many (non-trivial) scoring rules, including the popular *r-approval*, *plurality*, *veto*, *Borda count* and so on. In the case of r-approval, $\alpha = (\underbrace{1, 1, \ldots, 1}_{r}, \underbrace{0, 0, \ldots, 0}_{m-r})$. In the case of plurality, $\alpha = (1, 0, \ldots, 0)$. In the case of veto, $\alpha = (1, 1, \ldots, 1, 0)$. It is clear that plurality and veto are special cases of the scoring rule of r-approval.

Weights of voters. Voters can have different weights. Let $w_j \in \mathbb{N}$ be the weight of voter v_j. In a weighted election, the total score of a candidate is the *weighted sum* of the scores a candidate receives from the voters. For example, candidate c_i receives a score $w_j \cdot \alpha_{\tau_j^{-1}(i)}$ from voter v_j.

By re-indexing all of the candidates, we can set, without loss of generality, c_m as the winner in the absence of bribery.

Adversarial models. We consider an attacker that does not belong to $\mathcal{C} \cup \mathcal{V}$ but attempts to manipulate the election by bribing some voters. Suppose voter v_j has a *bribing price* p_j^b, meaning that v_j, upon receiving a bribery of amount p_j^b from the attacker, will change its preference list to the list given by the attacker. The attacker has a total budget B. As in the bribery problem, we also consider two kinds of attackers:

- *Constructive attacker*: This attacker attempts to make a designated candidate win the election, meaning that the designated candidate is the only candidate who gets the highest score.
- *Destructive attacker*: This attacker attempts to make a designated candidate lose the election, meaning that there is another candidate that gets a strictly higher score than the designated candidate does.

Protection. In the protection problem, voter v_j, upon receiving an award of amount p_j^a (or *awarding price*) from the defender, will always report its preference list faithfully and cannot be bribed. Note that p_j^a may have multiple interpretations, such as monetary award, economic incentives or the cost of iso-

lating voters from bribery. We say a voter v_j is *awarded* if v_j receives an award of p_j^a.

Problem statement. We formalize our problem as follows.

The constructive protection problem (i.e., protecting elections against constructive attackers):
Input: A set $\mathcal{C}$ of m candidates. A set $\mathcal{V}$ of n voters, each with a weight $w_j \in \mathbb{Z}_{>0}$, a preference list τ_j, an awarding price of $p_j^a \in \mathbb{Z}_{>0}$ and a bribing price of $p_j^b \in \mathbb{Z}_{>0}$. A scoring rule for selecting a single winner. A defender with a defense budget $F \in \mathbb{Z}_{\geq 0}$. An attacker with an attack budget $B \in \mathbb{Z}_{\geq 0}$ attempting to make candidate c_m win the election.
Output: Decide whether there exists a $\mathcal{V}_F \subseteq \mathcal{V}$ such that

- $\sum_{j:v_j \in \mathcal{V}_F} p_j^a \leq F$; and
- for *any* subset $\mathcal{V}_B \subseteq \mathcal{V} \setminus \mathcal{V}_F$ with $\sum_{j:v_j \in \mathcal{V}_B} p_j^b \leq B$, c_m does not get a strictly higher score than any other candidate despite the attacker bribing the voters belonging to $\mathcal{V}_B$ (i.e., bribing $\mathcal{V}_B$).

The destructive protection problem (i.e., protecting elections against destructive attackers):
Input: A set $\mathcal{C}$ of m candidates. A set $\mathcal{V}$ of n voters, each with a weight $w_j \in \mathbb{Z}_{>0}$, a preference list τ_j, an awarding price of $p_j^a \in \mathbb{Z}_{>0}$ and a bribing price of $p_j^b \in \mathbb{Z}_{>0}$. A scoring rule for selecting a single winner. Suppose c_m is the winner if no voter is bribed. A defender with a defense budget $F \in \mathbb{Z}_{\geq 0}$. An attacker with an attack budget $B \in \mathbb{Z}_{\geq 0}$ attempting to make c_m lose the election by making $c \in \mathcal{C} \setminus \{c_m\}$ get a strictly higher score than c_m does.
Output: Decide if there exists a $\mathcal{V}_F \subseteq \mathcal{V}$ such that

- $\sum_{j:v_j \in \mathcal{V}_F} p_j^a \leq F$; and
- for *any* subset $\mathcal{V}_B \subseteq \mathcal{V} \setminus \mathcal{V}_F$ such that $\sum_{j:v_j \in \mathcal{V}_B} p_j^b \leq B$, no candidate $c \in \mathcal{C} \setminus \{c_m\}$ can get a strictly higher score than c_m does despite the attacker bribing $\mathcal{V}_B$.

Further terminology and notations. We denote by $W(c_i)$ the total score obtained by candidate c_i in the absence of bribery (i.e., no voter is bribed). If the defender can select $\mathcal{V}_F$ such that no constructive or destructive attacker can succeed, we say the defender succeeds. We call our problem as the (constructive or destructive) *weighted-$-protection* problem, where "weighted" indicates that the voters are weighted and "$" indicates that arbitrary awarding and bribing prices are involved. In addition to investigating the general *weighted-$-protection* problem, we also investigate the following special cases of it:

- the *$-protection* problem with $w_j = 1$ for each j (i.e., the voters are not weighted);
- the *weighted-protection* problem with $p_j^a = p_j^b = 1$ for each j (i.e., voters are associated with the unit awarding price and the unit bribing price);

– the *unit-protection* problem with $w_j = p_j^a = p_j^b = 1$ for each j (i.e., voters are not weighted, and are associated with the unit awarding price and the unit bribing price).
– the *symmetric protection* problem with $p_j^a = p_j^b$ for each j (i.e., the awarding price and the bribing price are always the same), while noting that different voters may have different prices.

3 The Case of Constantly Many Candidates

3.1 The Weighted-$-Protection Problem

The goal of this subsection is to prove the following theorem.

Theorem 1. *For any non-trivial scoring rule, both the* constructive and destructive weighted-$-protection problem*, is Σ_2^p-complete.*

The theorem follows from Lemma 1 and Lemma 2 below, which shows the Σ_2^p membership and Σ_2^p-hardness, respectively.

Lemma 1. *For any non-trivial scoring rule, both the constructive and destructive weighted-$-protection problems are in Σ_2^p.*

Lemma 2. *For any non-trivial scoring rule, both the constructive and destructive weighted-$-protection problems are both Σ_2^p-hard even if there are only $m = 2$ candidates.*

Proof sketch. The proof of Lemma 2 follows from De-Negre (DNeg) variant of bi-level knapsack problem, which is proved to be Σ_2^p-hard by Caprara et al. [3]. We give a brief explanation. In this De-Negre variant, there are an adversary and a packer. The adversary has a reserving budget $\bar{F}$ and the packer has a packing budget $\bar{B}$. There is a set of n items, each having a price $\bar{p}_j^a$ to the adversary, a price $\bar{p}_j^b$ to the packer, and a weight $\bar{w}_j = \bar{p}_j^b$ to both the adversary and the packer. The adversary first reserves a subset of items whose total price is no more than $\bar{F}$. Then the packer solves the knapsack problem with respect to the remaining items that are not reserved; i.e., the packer will select a subset of the remaining items whose total price is no more than $\bar{B}$ but their total weight is maximized. The De-Negre variant asks if the adversary can reserve a proper subset of items such that the total weight of the unreserved items that are selected by the packer is no more than some parameter W. The De-Negre variant is similar to the weighted-$-protection protection problem, because we can view the defender and attacker in the protection problem respectively as the adversary and packer in the bi-level knapsack problem. In the case of a single-winner election with $m = 2$ candidates, the goal of the defender is to assure that the constructive attacker cannot make the loser get a strictly higher score than the winner by bribing. This is essentially the same as ensuring that the constructive attacker cannot bribe a subset of non-awarded voters whose total weight is higher than a certain threshold, which is the same as the bi-level knapsack problem. □

3.2 The Weighted-Protection Problem

This is a special case of the weighted-$-protection problem when $p_j^a = p_j^b = 1$.

The following theorem used by Faliszewski et al. [8] was originally proved for another problem. In our context, $F = 0$ and thus $\mathcal{V}_F = \emptyset$, it is NP-hard to decide if the constructive attacker can succeed or, equivalently, if the defender *cannot* succeed. Hence, it is coNP-hard to decide if the defender can succeed and Theorem 2 follows.

Theorem 2. (By Faliszewski et al. [8]) *If m is a constant, the constructive weighted-protection problem is coNP-hard for any scoring rule that $\alpha_2, \alpha_3, \ldots, \alpha_m$ are not all equal (i.e., it does not hold that $\alpha_2 = \alpha_3 = \ldots = \alpha_m$).*

In contrast, the destructive version is easy. Using the fact that m, the number of candidates, is a constant, we can prove the following Theorem 3 through suitable enumerations.

Theorem 3. *If m is a constant, then the destructive weighted-protection problem is in P for any scoring rule.*

3.3 The $-Protection Problem

This is the special case of the protection problem with $w_j = 1$ for every j. The following two theorems illustrate the significant difference (in terms of complexity) between the general problem and its special case with symmetricity (i.e., $p_j^a = p_j^b$).

Theorem 4. *For constant m and any non-trivial scoring rule, both the constructive and destructive $-protection problems are NP-complete.*

Theorem 5. *For constant m, both destructive and constructive symmetric $-protection problems are in P for any scoring rule.*

4 The Case of Arbitrarily Many Candidates

4.1 The Case of Constructive Attacker

The following Theorem 6 shows Σ_2^p-hardness for the most special cases of the constructive weighted-$-protection problem, namely $w_j = p_j^a = p_j^b = 1$ (unit-protection). It thus implies readily the Σ_2^p-hardness for the more general constructive $-protection and constructive weighted-protection.

Theorem 6. *For arbitrary m, the r-approval constructive unit-protection problem is Σ_2^p-complete.*

Membership in Σ_2^p follows directly from Lemma 1. To prove Theorem 6, it suffices to show the following.

Lemma 3. *For arbitrary m, the r-approval constructive unit-protection problem is Σ_2^p-hard even if $r = 4$.*

To prove Lemma 3, we reduce from a variant of the $\exists\forall$ 3 dimensional matching problem (or $\exists\forall$3DM), which is called $\exists\forall$3DM$'$ and defined below. The classical $\exists\forall$ 3DM is Σ_2^p-hard proved by Mcloughlin [16]. By leveraging the proof by Mcloughlin [16], we can show the Σ_2^p-hardness of the $\exists\forall$3DM$'$ problem.

$\exists\forall$3DM$'$: Given a parameter t, three disjoint sets of elements W, X, Y of the same cardinality, and two disjoint subsets M_1 and M_2 of $W \times X \times Y$ such that M_1 contains each element of $W \cup X \cup Y$ at most once. Does there exist a subset $U_1 \subseteq M_1$ such that $|U_1| = t$ and for any $U_2 \subseteq M_2$, $U_1 \cup U_2$ is *not* a *perfect matching* (where a *perfect matching* is a subset of triples in which every element of $W \cup X \cup Y$ appears exactly once)?

Proof (Proof of Lemma 3). Given an arbitrary instance of $\exists\forall$ 3DM$'$, we construct an instance of the constructive unit-protection problem in r-approval election as follows. Recall that $r = 4$ and thus every voter votes for 4 candidates.

Suppose $|W| = |X| = |Y| = n$, $|M_1| = m_1$, $|M_2| = m_2$.

There are $3n + 2$ key candidates, including:

- $3n$ key candidates, each corresponding to one distinct element of $W \cup X \cup Y$ and we call them element candidates. The score of every element candidate is $n + \xi$;
- one key candidate called leading candidate, whose total score is $n + t + \xi - 1$;
- one key candidate called designated candidate, whose total score is ξ.

Here ξ is some sufficiently large integer, e.g., we can choose $\xi = (m_1 + m_2)n$. Besides key candidates, there are also many dummy candidates, each of score either 1 or $m_1 - t + 1$. The number of dummy candidates will be determined later.

There are $m_1 + m_2(m_1 - t + 1)$ key voters, including:

- m_1 key voters, each corresponding to a distinct triple in M_1 and we call them M_1-voters. For each $(w_i, x_j, y_k) \in M_1$, the corresponding voter votes for the 3 candidates corresponding to elements w_i, x_j, y_k together with the leading candidate;
- $m_2 \cdot (m_1 - t + 1)$ key voters, each distinct triple in M_2 corresponds to exactly $m_1 - t + 1$ voters and we call them M_2-voters. For every $(w_i, x_j, y_k) \in M_2$, each of its $m_1 - t + 1$ corresponding voters vote for the 3 candidates corresponding to elements w_i, x_j, y_k together with one distinct dummy candidate. Since the $m_1 - t + 1$ voters are identical, we can view them as $m_1 - t + 1$ copies, i.e., every M_2-voter has $m_1 - t + 1$ copies.

Besides key voters, there are also sufficiently many dummy voters. Each dummy voter votes for exactly one key candidate and 3 distinct dummy candidates. Dummy voters and dummy candidates are used to make sure that the score of key candidates are exactly as we have described. More precisely, if we only count the scores of key candidates contributed by key voters, then the element candidate corresponding to $z \in W \cup X \cup Y$ has a score of $d(z) = d_1(z) + (m_1 -$

$t+1)d_2(z)$ where $d_i(z)$ is the number of occurrences of z in the triple set M_i for $i = 1, 2$, and the leading candidate has a score of m_1. Hence, there are exactly $n+\xi-d(z)$ dummy voters who vote for the element candidate corresponding to z, and $n+t+\xi-1-m_1$ dummy voters who vote for the leading candidate.

Overall, we create $\sum_{z\in W\cup X\cup Y}(n+\xi-d(z))+n+t+\xi-m_1-1$ dummy voters, and $3\sum_{z\in W\cup X\cup Y}(n+\xi-d(z))+3(n+t+\xi-m_1-1)+m_2$ dummy candidates.

As the leading candidate is the current winner, the constructive unit-protection problem asks whether the election can be protected against an attacker attempting to make the designated candidate win. The defense budget is $F = m_1-t$ and the attack budget is $B = n$. In the following we show that the defender succeeds if and only if the given $\exists\forall$ 3DM$'$ instance admits a feasible solution U_1.

"Yes" Instance of $\exists\forall$ 3DM$'$ $\rightarrow$ "Yes" Instance of Constructive Unit-Protection. Suppose the instance of $\exists\forall$ 3DM$'$ admits a feasible solution U_1, we show that the answer for constructive unit-protection problem is "Yes".

Recall that each M_1-voter corresponds to a distinct triple (w_i, x_j, y_k) in M_1 and votes for 4 candidates – the leading candidate and the three candidates corresponding to w_i, x_j, y_k. We do *not* award M_1-voters corresponding to the triples in U_1, but award all of the remaining M_1-voters. The resulting cost is exactly $F = m_1 - t$. In what follows we show that after awarding voters this way, the attacker cannot make the designated candidate win.

Suppose on the contrary, the attacker can make the designated candidate win by bribing $\alpha \leq t$ voters among the M_1-voters, $\beta \leq m_2$ voters among the M_2-voters, and γ dummy voters. We claim that the following inequalities hold:

$$\alpha+\beta+\gamma \leq n \tag{1a}$$

$$4\alpha+3\beta+\gamma \geq 3n+t \tag{1b}$$

Inequality (1a) follows from the fact that the attack budget is n and the attacker can bribe at most n voters. Inequality (1b) holds because of the following. Given that a candidate can get at most one score from each voter and that the attacker can bribe at most n voters, bribing voters can make the designated candidate obtain a score at most $n+\xi$. Hence, the score of each key candidate other than the designated one should be at most $n+\xi-1$. Recall that without bribery, each of the $3n$ element candidate has a score of $n+\xi$ and the leading candidate has a score of $n+t+\xi-1$. Hence, the attacker should decrease at least 1 score from each element candidate and t scores from the leading candidate, leading to a total score of $3n+t$. Note that an M_1-voter contributes 1 score to 4 key candidates, therefore it contributes in total a score of 4 to the key candidates. Similarly an M_2-voter contributes a score of 3, and a dummy voter contributes a score of 1 to the key candidates. Therefore, by bribing (for example) an M_1-voter, the total score of all the element candidates and the leading candidate can decrease by at most 4. Thus, inequality (1b) holds.

In the following we derive a contradiction based on Inequalities (1a) and (1b). By plugging $\gamma \leq n-\alpha-\beta$ into Inequality (1b), we have $3\alpha+2\beta \geq 2n+t$. Since $\beta \leq n-\alpha$, we have $3\alpha+2\beta \leq \alpha+2n \leq 2n+t$. Hence, $3\alpha+2\beta = \alpha+2n = 2n+t$,

and we have $\alpha = t$ and $\beta = n - t$. Note that the defender has awarded every M_1-voter except the ones corresponding to U_1, where $|U_1| = t$. Hence, every voter corresponding to the triples in U_1 is bribed. Furthermore, as Inequality (1b) is tight, bribing voters makes the designated candidate have a score of $n + \xi$, while making each of the other key candidates have a score of $n + \xi - 1$. This means that the score of each element candidate decreases exactly by 1. Hence, the attacker has selected a subset of M_2-voters such that together with the M_1-voters corresponding to triples in U_1, these voters contribute exactly a score of 1 to every element candidate. Let U_2 be the set of triples to which the bribed M_2-voters correspond, then $U_1 \cup U_2$ forms a 3-dimensional matching, which is a contradiction to the fact that U_1 is a feasible solution to the $\exists\forall$ 3DM$'$ instance. Thus, the attacker cannot make the designated candidate win and the answer for the constructive unit-protection problem is "Yes".

"No" Instance of $\exists\forall$ 3DM$'$ $\rightarrow$ "No" Instance of Constructive Unit-Protection. Suppose for any $U_1 \subseteq M_1$, $|U_1| = t$ there exists $U_2 \in M_2$ such that $U_1 \cup U_2$ is a perfect matching, we show that the answer to the constructive unit-protection problem is "No". Consider an arbitrary set of voters awarded by the defender. Among the awarded voters, let H be the set of triples that corresponds to the awarded M_1-voters. As $|H| \leq m_1 - t$, $|M_1 \setminus H| \geq t$. We select an arbitrary subset $H_1 \subseteq M_1 \setminus H$ such that $|H_1| = t$. There exists some $H_2 \subseteq M_2$ such that $H_1 \cup H_2$ is a perfect matching, and we let the attacker bribe the set of voters corresponding to triples in $H_1 \cup H_2$. Note that this is always possible as every M_2-voter has $m_1 - t + 1$ copies, so no matter which M_2-voters are awarded the briber can always select one M_2-voter corresponding to each triple in H_2. It is easy to see that by bribing these voters, the score of every element candidate decreases by 1, and the score of the leading voter decreases by t. Meanwhile, let each bribed voter vote for the designated candidate and three distinct dummy candidates, then the designated candidate has a score of $n + \xi$ and becomes a winner, i.e., the answer to the constructive unit-protection problem is "No". □

Remark. The proof of Lemma 3 can be easily modified to prove the Σ_2^p-hardness of r-approval constructive unit-protection problem for any fixed $r \geq 4$. Specifically, we can make the same reduction, and add dummy candidates such that every voter additionally votes for exactly $r - 4$ distinct dummy candidates.

4.2 The Case of Destructive Attacker

Theorem 7. *Both r-approval destructive weighted-protection and r-approval (symmetric) \$-protection problems are NP-complete.*

The proof of Theorem 7 is based on a crucial observation of the equivalence between the destructive weighted-\$-protection problem (under an arbitrary scoring rule) and the *minmax vector addition problem* we introduce (see Appendix 7.1). The full proof of Theorem 7 can be found in Appendix 7.6.

5 Summary of Results

The preceding characterization of the computational complexity of the protection problem in various settings is summarized in Table 1.

Table 1. Summary of results for *single-winner* election under the *r-approval* scoring rule: "Symmetric" means $p_j^a = p_j^b$ for every j and "asymmetric" means otherwise; hardness results that are proved for the case with only two candidates (i.e., $m = 2$) are marked with a "$\diamond$" (Note that when $m = 2$, the 1-approval rule is the same as the plurality, veto or Borda scoring rule. It can be shown that with a slight modification, the hardness results hold for any *non-trivial* scoring rule); algorithmic results (marked with a "P") hold for arbitrary scoring rules; the complexity of the protection problem against a destructive attacker with $w_j = p_j^a = p_j^b = 1$ remains open; for most variants of the protection problem against a constructive attacker, we only provide hardness results and we do not know yet whether or not they belong to the class of coNP-complete or Σ_2^p-complete proble.

# of candidates	Model parameters	Destructive attacker	Constructive attacker
constant	Weighted, Priced, Asymmetric	Σ_2^p-complete $\diamond$ (Thm 1)	Σ_2^p-complete $\diamond$ (Thm 1)
	Weighted, $p_j^a = p_j^b = 1$	P (Thm 3)	coNP-hard (Thm 2)
	$w_j = 1$, Priced, Asymmetric	NP-complete $\diamond$ (Thm 4)	NP-complete $\diamond$ (Thm 4)
	$w_j = 1$, Priced, Symmetric	P (Thm 5)	P (Thm 5)
	$w_j = 1$, $p_j^a = p_j^b = 1$	P (Thm 5)	P (Thm 5)
arbitrary	Weighted, Priced, Asymmetric	Σ_2^p-complete $\diamond$ (Thm 1)	Σ_2^p-complete $\diamond$ (Thm 1)
	Weighted, $p_j^a = p_j^b = 1$	NP-complete (Thm 7)	Σ_2^p-hard (Thm 6)
	$w_j = 1$, Priced, Asymmetric	NP-complete (Thm 7)	Σ_2^p-hard (Thm 6)
	$w_j = 1$, Priced, Symmetric	NP-complete (Thm 7)	Σ_2^p-hard (Thm 6)
	$w_j = 1$, $p_j^a = p_j^b = 1$	?	Σ_2^p-hard (Thm 6)

We remark three natural open problems for future research. One is the complexity of the destructive protection problem with $w_j = p_j^a = p_j^b = 1$. It is not clear whether the problem is in P or is NP-complete. Another is the constructive protection problem with $p_j^a = p_j^b = 1$ and arbitrary voter weights. We only show its coNP-hardness, it is not clear whether or not this problem is coNP-complete. The third problem is the complexity of r-approval constructive unit-protection problem when $r = \{1, 2, 3\}$ as our hardness proof only holds when $r \geq 4$.

6 Conclusion

We introduced the protection problem and characterized its computational complexity. We showed that the problem, in general, is Σ_2^p-complete, and identified settings in which the problem becomes easier. Moreover, we showed the protection problem in some parameter settings is polynomial-time solvable, suggesting that these parameter settings can be used for real-work election applications.

In addition to the open problems mentioned in Section 5, the following are also worth investigating. First, our hardness results would motivate the study of approximation or FPT (fixed parameter tractable) algorithms for the protection problem. Note that even polynomial time approximation schemes can exist for Σ_2^p-hard problems (see, e.g., By Caparara et al. [3]). It is thus desirable that a similar result can be obtained for some variants of the protection problem. Second, how effective is this approach when applied towards the problem of defending against other types of attackers that can, e.g., add or delete votes? Third, much research remains to be done in extending the protection problem to accommodate other scoring rules such as Borda and Copeland.

7 Appendix

7.1 Destructive Weighted-Protection – an Equivalent Formulation

We provide an equivalent formulation of the destructive weighted-protection problem under any scoring rule $\alpha = (\alpha_1, \cdots, \alpha_m)$, which will be very useful for several proofs throughout this paper.

The minmax vector addition problem:
Input: A vector $\boldsymbol{\Lambda} = (\Lambda(c_1), \Lambda(c_2), \ldots, \Lambda(c_{m-1}))$ where $\Lambda(c_i)$ is the score of c_i in the absence of bribery. An $(m-1)$-vector $\boldsymbol{\Delta}_j = (\Delta_{1j}, \Delta_{2j}, \ldots, \Delta_{(m-1),j})$ for each voter v_j where $\Delta_{ij} = \alpha_1 - \alpha_{\tau_j^{-1}(i)} + \alpha_{\tau_j^{-1}(m)} - \alpha_m$. Awarding price p_j^a and bribing price p_j^b for voter v_j, $j = 1, 2, \ldots, n$. Defense budget F and attack budget B.
Output: Decide if there exists a subset $\mathcal{V}_F \subseteq \mathcal{V}$ such that

- $\sum_{j:v_j \in \in \mathcal{V}_F} p_j^a \leq F$; and
- For *any* subset $\mathcal{V}_B \subseteq \mathcal{V} \setminus \mathcal{V}_F$ with $\sum_{j:v_j \in \mathcal{V}_B} p_j^b \leq B$, it holds that

$$\left\| \boldsymbol{\Lambda} + \sum_{j:v_j \in \mathcal{V}_B} w_j \boldsymbol{\Delta}_j \right\|_\infty \leq \Lambda(c_m),$$

where $\|\cdot\|_\infty$ is the infinity norm (i.e., the maximal absolute value among the $m-1$ coordinates).

Lemma 4. *The answer to the destructive weighted-$-protection problem is "Yes" if and only if the answer to the corresponding minmax vector addition problem is "Yes".*

Proof. **A "Yes" Instance of Minmax Vector Addition → A "Yes" Instance of Destructive Weighted-$-Protection.** Suppose the answer to the minmax vector addition problem is "Yes." Then, there exists some $\mathcal{V}_F^* \subseteq \mathcal{V}$ such that for any $\mathcal{V}_B \subseteq \mathcal{V} \setminus \mathcal{V}_F^*$ with $\sum_{j:v_j \in \mathcal{V}_B} p_j^b \leq B$, it holds that

$$\left\| \boldsymbol{\Lambda} + \sum_{j:v_j \in \mathcal{V}_B} w_j \boldsymbol{\Delta}_j \right\|_\infty \leq \Lambda(c_m). \tag{2}$$

For showing a contradiction, suppose the answer for the destructive weighted-$-protection problem is "No". In this case, even if the defender awards the voters in $\mathcal{V}_F^*$, the attacker can still make c_m lose by bribing some subset $\mathcal{V}_B^*$ of voters. Note that if c_m does not win, there must exist some other candidate, say, c_i, who gets a strictly higher score than c_m after the attacker bribes some voters. Let us compare their scores before and after bribing voters. Before bribing voters, the scores of c_i and c_m are $\Lambda(c_i)$ and $\Lambda(c_m)$, respectively. Recall that a candidate c_k is at the position of $\tau_j^{-1}(k)$ on the preference list of v_j, therefore any $v_j \in \mathcal{V}_B^*$

contributes a score of $\alpha_{\tau_j^{-1}(i)}$ to $\Lambda(c_i)$, and contributes a score of $\alpha_{\tau_j^{-1}(m)}$ to $\Lambda(c_m)$. After bribing voters, the preference list of v_j is changed, but regardless of the change, v_j contributes at least α_m to c_m and at most α_1 to c_i. Let the scores of c_i and c_m after bribing voters be $\Lambda'(c_i)$ and $\Lambda'(c_m)$, respectively. Then, it follows that

$$\Lambda'(c_i) \leq \Lambda(c_i) + \sum_{j:v_j \in \mathcal{V}_B^*} w_j(\alpha_1 - \alpha_{\tau_j^{-1}(i)})$$
$$\Lambda'(c_m) \geq \Lambda(c_m) + \sum_{j:v_j \in \mathcal{V}_B^*} w_j(\alpha_m - \alpha_{\tau_j^{-1}(m))}.$$

Since $\Lambda'(c_i) > \Lambda'(c_m)$, we have

$$\Lambda(c_i) + \sum_{j:v_j \in \mathcal{V}_B^*} w_j(\alpha_1 - \alpha_{\tau_j^{-1}(i)}) > \Lambda(c_m) + \sum_{j:v_j \in \mathcal{V}_B^*} w_j(\alpha_m - \alpha_{\tau_j^{-1}(m)}),$$

that is,

$$\Lambda(c_i) + \sum_{j:v_j \in \mathcal{V}_B^*} w_j \Delta_{ij} > \Lambda(c_m),$$

which contradicts Eq. (2). Thus, the answer to the destructive weighted-$-protection problem is "Yes".

A "Yes" Instance of Destructive Weighted-$-Protection → A "Yes" Instance of Minmax Vector Addition. Suppose the answer to the destructive weighted-$-protection problem is "Yes" by awarding the voters in $\mathcal{V}_F^*$. We show that the answer to the corresponding instance of minmax vector addition problem is "Yes". Suppose on the contrary the answer is "No." Then, for $\mathcal{V}_F^*$ there exists some $\mathcal{V}_B^* \subseteq \mathcal{V} \setminus \mathcal{V}_F^*$ such that

$$\left\| \boldsymbol{\Lambda} + \sum_{j:v_j \in \mathcal{V}_B^*} w_j \boldsymbol{\Delta}_j \right\|_\infty > \Lambda(c_m).$$

Consequently, there must exist some $1 \leq i \leq m-1$ such that $\Lambda(c_i) + \sum_{j:v_j \in \mathcal{V}_B^*} \Delta_{ij} > \Lambda(c_m)$. By plugging in Δ_{ij}, we have

$$\Lambda(c_i) + \sum_{j:v_j \in \mathcal{V}_B^*} w_j(\alpha_1 - \alpha_{\tau_j^{-1}(i)}) > \Lambda(c_m) + \sum_{j:v_j \in \mathcal{V}_B^*} w_j(\alpha_m - \alpha_{\tau_j^{-1}(m)}).$$

This means that if the defender awards the voters in $\mathcal{V}_F^*$, then the attacker can bribe the voters in $\mathcal{V}_B^*$ to change their preference lists such that for any $v_j \in \mathcal{V}_B^*$, candidate c_i is on top of the list and c_m is at bottom of the list. By doing this, c_i gets a strictly higher score than c_m. This contradicts the fact that the answer to the destructive weighted-$-protection problem is "Yes". Hence, the answer to the minmax vector addition problem is "Yes". □

7.2 Voter Dominance and Preliminary Observations

Let S_m be the set of permutations over $\{c_1, c_2, \ldots, c_m\}$. Each element of S_m can be a preference list. Let $\mathcal{V}^h \subseteq \mathcal{V}$ be the set of voters whose preference list is the h-th element of S_m.

For two voters v_j and $v_{j'}$, we say v_j *dominates* $v_{j'}$ (or $v_{j'}$ is *dominated* by v_j), denoted by $v_{j'} \prec v_j$, if any of the following two conditions hold: (i) The following holds and at least one of the inequalities is strict:

$$(\tau_j = \tau_{j'}) \ \wedge \ (w_j \geq w_{j'}) \ \wedge \ (p_j^a \leq p_{j'}^a) \ \wedge \ (p_j^b \leq p_{j'}^b).$$

(ii) The following holds:

$$(\tau_j = \tau_{j'}) \wedge (w_j = w_{j'}) \wedge (p_j^a = p_{j'}^a) \wedge (p_j^b = p_{j'}^b) \wedge (j' < j).$$

Note that the domination relation is only defined between the voters who have the *same* preference. Intuitively, if $v_{j'} \prec v_j$, then v_j is more "important" than $v_{j'}$ because v_j has a greater weight but is "cheaper" to bribe or award (i.e., more valuable to both the attacker and the defender).

We have the following lemmas.

Lemma 5. *Consider the destructive weighted-$-protection problem with $\mathcal{V}_F \subseteq \mathcal{V}$ being the set of awarded voters. Suppose the attacker can succeed by bribing a subset $\mathcal{V}_B \subseteq \mathcal{V} \setminus \mathcal{V}_F$ of voters. If $v_{j'} \prec v_j$, $v_{j'} \in \mathcal{V}_B$ and $v_j \notin (\mathcal{V}_F \cup \mathcal{V}_B)$, then the attacker can succeed by bribing $(\mathcal{V}_B \setminus \{v_{j'}\}) \cup \{v_j\}$.*

Towards the proof, we need Lemma 4, which states that the destructive weighted-$-protection problem is equivalent to another problem called minmax vector addition. The reader may refer to Section 4.2 for the definition of this problem. The following observation follows directly from the definition of Δ_{ij} which is included in the definition of minmax vector addition.

Observation 1 *If $v_{j'} \prec v_j$, then $\Delta_{ij'} \leq \Delta_{ij}$ for $1 \leq i \leq m-1$.*

Assuming Lemma 4, we can prove Lemma 5 as follows.

Proof (Proof of Lemma 5). We prove the lemma by applying an exchange argument to the minmax vector addition problem, which is equivalent to the destructive weighted-$-protection by Lemma 4. Suppose by bribing voters in $\mathcal{V}_B$ the destructive attacker can make c_m lose. Then, it follows that $\sum_{j:v_j \in \mathcal{V}_B} p_j^b \leq B$ and

$$||\boldsymbol{\Lambda} + \sum_{j:j \in \mathcal{V}_B} w_j \boldsymbol{\Delta}_j||_\infty > \Lambda(c_m).$$

As v_j dominates $v_{j'}$, $\Delta_{ij} \geq \Delta_{ij'}$ and $w_j \leq w_{j'}$. Hence,

$$\sum_{j:j \in (\mathcal{V}_B \setminus \{v_{j'}\}) \cup \{v_j\}} p_j^b \leq B$$

and

$$||\boldsymbol{\Lambda} + \sum_{j:j\in(\mathcal{V}_B\setminus\{v_{j'}\})\cup\{v_j\}} w_j\boldsymbol{\Delta}_j||_\infty > \Lambda(c_m).$$

That is, the briber can also win by bribing voters in $(\mathcal{V}_B \setminus \{v_{j'}\}) \cup \{v_j\}$. □

Lemma 6. *Consider the destructive weighted-$-protection problem. Suppose the defender succeeds by awarding a subset $\mathcal{V}_F \subseteq \mathcal{V}$ of voters. If $v_{j'} \prec v_j$, $v_{j'} \in \mathcal{V}_F$ and $v_j \notin \mathcal{V}_F$, then the defender can succeed by awarding $(\mathcal{V}_F \setminus \{v_{j'}\}) \cup \{v_j\}$.*

Proof. We again use an exchange argument to the minmax vector addition problem. Let $\mathcal{V}_A = \mathcal{V}_F \setminus \{v_{j'}\}$. Suppose on the contrary that the defender cannot win by fixing voters in $\mathcal{V}_A \cup \{v_j\}$, then there exists some $\mathcal{V}_B \subseteq \mathcal{V} \setminus (\mathcal{V}_A \cup \{v_j\})$ such that $\sum_{j:v_j\in\mathcal{V}_B} p_j^b \leq B$ and

$$||\boldsymbol{\Lambda} + \sum_{j:j\in\mathcal{V}_B} w_j\boldsymbol{\Delta}_j||_\infty > \Lambda(c_m).$$

We argue that the defender cannot win either by fixing voters in $\mathcal{V}_A \cup \{v_{j'}\}$, which is a contradiction. Suppose the defender fixes voters in $\mathcal{V}_A \cup \{v_{j'}\}$. There are two possibilities. If $v_{j'} \notin \mathcal{V}_B$, then we let the briber bribe voters in $\mathcal{V}_B$. It is obvious that the briber can win. Otherwise, $v_{j'} \in \mathcal{V}_B$, then we let the briber bribe voters in $(\mathcal{V}_B \setminus \{v_{j'}\}) \cup \{v_j\}$. Since v_j dominates $v_{j'}$, we have $\sum_{j:v_j\in(\mathcal{V}_B\setminus\{v_{j'}\})\cup\{v_j\}} p_j^b \leq B$ and

$$||\boldsymbol{\Lambda} + \sum_{j:j\in(\mathcal{V}_B\setminus\{v_{j'}\})\cup\{v_j\}} w_j\boldsymbol{\Delta}_j||_\infty > \Lambda(c_m).$$

Hence, the lemma is true. □

We say $\bar{\mathcal{V}} \subseteq \mathcal{V}$ is maximal (with respect to $\mathcal{V}$) if for any $\bar{v} \in \bar{\mathcal{V}}$, there is no $v \in \mathcal{V} \setminus \bar{\mathcal{V}}$ that can dominate $\bar{v}$. That is, $\bar{\mathcal{V}}$ contains the most important voters. The following corollary follows directly from the preceding two lemmas.

Corollary 1 *Consider the destructive weighted-$-protection problem. Without loss of generality, we can assume that $\mathcal{V}_F$ is maximal with respect to $\mathcal{V}$ and $\mathcal{V}_B$ is maximal with respect to $\mathcal{V} \setminus \mathcal{V}_F$.*

Unfortunately, Corollary 1 does not hold for the constructive weighted-$-protection problem, which is significantly different from the destructive version of the protection problem in terms of computational complexity. Nevertheless, similar results hold for the unweighted constructive problem.

Lemma 7. *Given $\mathcal{V}_F \subseteq \mathcal{V}$ as the set of fixed voters in the constructive $-bribery-protection problem, suppose a briber can make c_i win by bribing a subset $\mathcal{V}_B \subseteq \mathcal{V} \setminus \mathcal{V}_F$ of voters. If $v_{j'} \prec v_j$, $v_{j'} \in \mathcal{V}_B$ and $v_j \notin (\mathcal{V}_F \cup \mathcal{V}_B)$, then the briber can also win by bribing voters in $(\mathcal{V}_B \setminus \{v_{j'}\}) \cup \{v_j\}$.*

Proof. Again, we prove by an exchange argument. Suppose by bribing voters in $\mathcal{V}_B$ the constructive briber can make c_i win. Now we consider the following procedure: we change the preference list of v_j into the same one as that of $v_{j'}$, and meanwhile, restore the preference list of $v_{j'}$ to the original one. As voters have the same weight, this procedure does not change the total score of every candidate, and c_i is thus still the winner. Furthermore, this procedure is equivalent as we bribe $(\mathcal{V}_B \setminus \{v_{j'}\}) \cup \{v_j\}$. Since v_j dominates $v_{j'}$, the total cost of bribing $(\mathcal{V}_B \setminus \{v_{j'}\}) \cup \{v_j\}$ is no more than that of bribing $\mathcal{V}_B$. Hence, the lemma is true. □

Lemma 8. *In the constructive \$-bribery-protection problem, suppose the defender can win by fixing a subset $\mathcal{V}_F \subseteq \mathcal{V}$ of voters. If $v_{j'} \prec v_j$, $v_{j'} \in \mathcal{V}_F$ and $v_j \notin \mathcal{V}_F$, then the defender can also win by fixing voters in $(\mathcal{V}_F \setminus \{v_{j'}\}) \cup \{v_j\}$.*

Proof. Suppose on the contrary that the defender cannot win by fixing voters in $\mathcal{V}'_F = (\mathcal{V}_F \setminus \{v_{j'}\}) \cup \{v_j\}$. Then the constructive briber can win by bribing voters in some subset $\mathcal{V}_B \subseteq \mathcal{V} \setminus \mathcal{V}'_F$. There are two possibilities. If $v_{j'} \notin \mathcal{V}_B$, then even if the defender fixes $\mathcal{V}_F$ the briber can still bribe $\mathcal{V}_B$ and make c_i win, which is a contradiction. Otherwise $v_{j'} \in \mathcal{V}_B$. If the defender fixes $\mathcal{V}_F$, we let the briber bribe $(\mathcal{V}_B \setminus \{v_{j'}\}) \cup \{v_j\}$. According to Lemma 7, if the briber can win by bribing $\mathcal{V}_B$, he/she can also win by bribing $(\mathcal{V}_B \setminus \{v_{j'}\}) \cup \{v_j\}$, again contradicting the fact that the defender can succeed by awarding $\mathcal{V}_F$. □

The above Lemmas implies the following.

Corollary 2 *Without loss of generality, we can assume that $\mathcal{V}_F$ is maximal with respect to $\mathcal{V}$, and $\mathcal{V}_B$ is maximal with respect to $\mathcal{V} \setminus \mathcal{V}_F$ in the constructive \$-bribery-protection problem.*

7.3 Proofs Omitted in Section 3.1

Recall that our goal is to prove Theorem 1.

Theorem 1. *For any non-trivial scoring rule, both the* constructive and destructive weighted-\$-protection problem*, is Σ_2^p-complete.*

We first show Σ_2^p-membership.

Lemma 1. *For any non-trivial scoring rule, both the constructive and destructive weighted-\$-protection problems are in Σ_2^p.*

For ease of proof, we use the following definition of Σ_2^p from [19] (see Theorem 3 therein).

Definition 1 (By Wrathall [19]) *Let Γ be a finite set of symbols (alphabet) and Γ^+ be the set of strings of symbols in Γ. Let $L \subseteq \Gamma^+$ be a language. $L \in \Sigma_2^p$ if and only if there exists polynomials ϕ_1, ϕ_2 and a language $L' \in P = \Sigma_0^p$ such that for all $x \in \Gamma^+$,*

$$x \in L \quad \text{if and only if} \quad (\exists y_1)_{\phi_1}(\forall y_2)_{\phi_2}[(x, y_1, y_2) \in L'],$$

where $(\exists y_1)_{\phi_1}(\forall y_2)_{\phi_2}[(x, y_1, y_2) \in L']$ *denotes*

$$(\exists y_1)(\forall y_2)[|y_1| \leq \phi_1(|x|) \text{ and if } |y_2| \leq \phi_2(|x|), (x, y_1, y_2) \in L'].$$

Proof (Proof of Lemma 1). Given an instance I of the constructive or destructive weighted-$-protection problem, we want to know if there exists a subset $\mathcal{V}_F$ such that $\sum_{j:v_j \in \mathcal{V}_F} p_j^a \leq F$ and for any subset $\mathcal{V}_B \subseteq \mathcal{V} \backslash \mathcal{V}_F$ with $\sum_{j:v_j \in \mathcal{V}_B} p_j^b \leq B$ and any preference list $\hat{\tau}_j$ for $v_j \in \mathcal{V}_B$, the following property $\mathcal{R}(I, \mathcal{V}_F, \mathcal{V}_B \cup \{\hat{\tau}_j | v_j \in \mathcal{V}_B\})$ is true: By bribing voter in $\mathcal{V}_B$ and change the preference list of each $v_j \in \mathcal{V}_B$ to $\hat{\tau}_j$, a constructive attacker cannot make candidate c_i win, or a destructive attacker cannot make c_m lose. It is easy to see that the property $\mathcal{R}(I, \mathcal{V}_F, \mathcal{V}_B \cup \{\hat{\tau}_j | v_j \in \mathcal{V}_B\})$ can be verified in polynomial time, therefore Lemma 1 is proved. □

Now we prove the Σ_2^p-hardness.

Lemma 2. *For any non-trivial scoring rule, both the constructive and destructive weighted-$-protection problems are both Σ_2^p-hard even if there are only $m = 2$ candidates.*

Note that in case of $m = 2$, destructive weighted-$-bribery-protection is equivalent to constructive weighted-$-bribery-protection. We prove the Lemma 2 for the protection problem under plurality. With slight modification the proof works for any non-trivial scoring protocol for two candidates.

We reduce from the De-Negre (DNeg) variant of bi-level knapsack problem, which is proved to be Σ_2^p-hard by Caprara et al. [3]. Before we describe the bi-level knapsack problem, we first introduce the classical knapsack problem, which is closely related. In the knapsack problem, given is some fixed budget $\bar{B}$ together with a set S of items, each having a price $\bar{p}_j^a$ and a weight $\bar{w}_j$. The goal is to select a subset of items whose total price is no more than the given budget and the total weight is maximized. We denote by $KP(S, \bar{B})$ the optimal objective value of the knapsack problem.

In the De-Negre (DNeg) variant of bi-level knapsack problem, there is an adversary and a packer. The adversary has a reserving budget $\bar{F}$ and the packer has a packing budget $\bar{B}$. There is a set of n items, each having a price $\bar{p}_j^a$ to the adversary, $\bar{p}_j^b$ to the packer and a weight $\bar{w}_j = \bar{p}_j^b$ (to both the adversary and the packer). The adversary first reserves a subset of items whose total prices is no more than $\bar{F}$. Then the packer solves the knapsack problem with respect to the remaining items that are not reserved, i.e., the packer will select a subset of remaining items whose total price is no more than $\bar{B}$ such that their total weight is maximized. The DNeg variant of the bi-level knapsack problem asks for a proper subset of items reserved by the adversary such that the total weight of items selected by the packer is minimized. More precisely, the problem can be formulated as a bi-level integer programming as follows.

The DNeg variant of bi-level knapsack problem:

$$\text{Minimize} \quad \sum_{j=1}^{n} \bar{p}_j^b y_j$$

$$s.t. \quad \sum_{j=1}^{n} \bar{p}_j^a x_j \leq \bar{F}$$

where $y_1, y_2, \cdots, y_n$ solves the following:

$$\text{Maximize} \quad \sum_{j=1}^{n} \bar{p}_j^b y_j$$

$$s.t. \quad \sum_{j=1}^{n} \bar{p}_j^b y_j \leq \bar{B}$$

$$x_j + y_j \leq 1 \qquad 1 \leq j \leq n$$

$$x_j, y_j \in \{0, 1\} \quad 1 \leq j \leq n$$

The decision version of the DNeg variant of bi-level knapsack problem asks whether there exists a feasible solution with the objective value at most W. The following lemma is due to Caprara et al. [3].

Lemma 9 (By Caprara et al. [3]). *The decision version of the DNeg variant of bi-level knapsack problem is Σ_2^p-complete.*

Based on the above lemma, we are able to prove Lemma 2.

Proof (Proof of Lemma 2). Given an arbitrary instance of the (decision version of) DNeg variant of bi-level knapsack problem, we construct an election instance as follows. There are $m = 2$ candidates. The defense and attack budgets are $F = \bar{F}$ and $B = \bar{B}$, respectively. There are $n + 2$ voters:

- n key voters $v_1, v_2, \cdots, v_n$ who vote for c_2, each having an awarding price $p_j^a = \bar{p}_j^a$, a bribing price $p_j^b = \bar{p}_j^b$, and a weight $w_j = \bar{p}_j^b$.
- one dummy voter v_{n+1} who votes for c_2 whose weight is $2W$, awarding price is $F + 1$ and bribing price is $B + 1$.
- one dummy voter v_{n+2} who votes for c_1 whose weight is $\sum_{j=1}^{n} \bar{p}_j^b$, awarding price is $F + 1$ and bribing price is $B + 1$.

Obviously c_2 is the original winner. We show in the following that the constructed election instance is secure if and only if the DNeg variant of bi-level knapsack problem admits a feasible solution with an objective value at most W.

Suppose the DNeg variant of bi-level knapsack problem admits a feasible solution with an objective value at most W, and let x_i^* be such a solution. As $\sum_{j=1}^{n} \bar{p}_j^a x_j^* = \sum_{j=1}^{n} p_j^a x_j^* \leq \bar{F} = F$, we let the defender award all the voters such that $x_j^* = 1$, i.e., let $\mathcal{V}_F = \{v_j | x_j^* = 1\}$. According to the fact that the objective value of the bi-level knapsack problem is at most W, and the fact that the two

dummy voters can never be bribed, it follows that the optimal objective value of the following knapsack problem is at most W:

$$\begin{aligned} \text{Maximize} \quad & \sum_{j:v_j \in \mathcal{V} \setminus \mathcal{V}_F} p_j^b y_j \\ \text{s.t.} \quad & \sum_{j \in \mathcal{V} \setminus \mathcal{V}_F} p_j^b y_j \leq B \end{aligned}$$

Thus, with a budget of B the the briber can never bribe key voters whose total weight is more than W. Note that originally the total weight of voters voting for c_2 is $2W + \sum_{j=1}^{n} p_j^b$, and the total weight of voters voting for c_1 is $\sum_{j=1}^{n} p_j^b$, the briber cannot succeed and the election is thus secure.

Suppose the election is secure. Then there exists some $\mathcal{V}_F$ such that we can say $\sum_{j:v_j \in \mathcal{V}_F} p_j^a \leq F$ and if voters in $\mathcal{V}_F$ are fixed, no briber can succeed. Note that the two dummy voters can never be protected nor bribed, whereas $\mathcal{V}_F \subseteq \{v_1, v_2, \cdots, v_n\}$. Thus, among voters in $\{v_1, v_2, \cdots, v_n\} \setminus \mathcal{V}_F$, within a budget of B the briber cannot bribe voters whose total weight is more than W. This is equivalent as saying that by setting $x_j = 1$ for $v_j \in \mathcal{V}_F$ and $x_j = 0$ otherwise, the knapsack problem for y_j does not admit a feasible solution with an objective value more than W. Hence, the objective value of the given DNeg variant of bi-level knapsack problem is at most W. □

7.4 Proofs Omitted in Section 3.2

Theorem 3. *If m is a constant, then the destructive weighted-protection problem is in P for any scoring rule.*

Proof. The theorem is proved by trying all different possible $\mathcal{V}_F$ and check whether the attacker can succeed for each of them. By Corollary 1, for voters having the same preference, $\mathcal{V}_F$ contains voters of the largest weights. Hence to determine $\mathcal{V}_F$, it suffices to know the number of voters having each preference in $\mathcal{V}_F$. There are at most $m!$ different preferences, and consequently at most $n^{m!}$ different kinds of $\mathcal{V}_F$, which is polynomial when m is constant. For each possible choice of $\mathcal{V}_F$, we check whether the attacker can succeed by trying all possible $\mathcal{V}_B$ and all possible ways of changing their preferences. Firstly, by Corollary 1, for voters in $\mathcal{V} \setminus \mathcal{V}_F$ that have the same preference list, $\mathcal{V}_B$ contains the ones of the largest weights, hence using a similar argument we know there are at most $n^{m!}$ different kinds of $\mathcal{V}_B$. Given $\mathcal{V}_F$ and $\mathcal{V}_B$, it remains to determine how the preference lists of voters in $\mathcal{V}_B$ should be changed. Note that we do not need to specify how the preference list is changed for each $v_j \in \mathcal{V}_B$. Instead, we only need to determine the number of voters in $\mathcal{V}_B$ that are changed to each preference list, which gives rise to at most $n^{m!}$ possibilities. Therefore, overall there are at most $n^{3m!}$ different possibilities regarding $\mathcal{V}_F$, $\mathcal{V}_B$ and how to alter the preference lists of voters in $\mathcal{V}_B$, which can be enumerated efficiently when m is a constant. □

We remark that an argument similar to the one we used to prove Theorem 3 was used by Faliszewsk et al. [8].

7.5 Proofs omitted in Section 3.3

Theorem 4. *For any non-trivial scoring rule, both the constructive and destructive $-protection problems are NP-complete.*

To prove Theorem 4, we first show the problem belongs to NP in Lemma 10 and then show its NP-hardness in Lemma 11.

Lemma 10. *For any scoring rule and arbitrary constant m, both the constructive and destructive $-protection problems are in NP.*

Proof. Note that to show the membership in NP, it suffices to show that given $\mathcal{V}_F$, we can determine in polynomial time whether the constructive/destructive attacker can succeed. Note that among voters of the same preference list in $\mathcal{V} \setminus \mathcal{V}_F$, $\mathcal{V}_B$ always contains the ones with the smallest bribing prices. Hence, similarly as the proof of Theorem 3, there are at most $n^{m!}$ different kinds of $\mathcal{V}_B$. Given $\mathcal{V}_B$, a similar argument as that of Theorem 3 shows that there are $n^{m!}$ different ways of altering the preference lists of voters in $\mathcal{V}_B$. Hence in $n^{2m!}$ time we can determine the whether the constructive/destructive attacker can succeed, which is polynomial if m is a constant. □

Lemma 11. *For any non-trivial scoring rule, both the constructive and destructive $-protection problems are NP-hard even if there are only 2 candidates.*

Again, in case of two candidates, the constructive and destructive variants are identical and it suffices to prove the theorem under the scoring rule of plurality.

Towards the proof, we need the following intermediate problems.

Balanced Partition: Given a set of positive integers $\{a_1, a_2, \cdots, a_{2n}\}$ where $a_1 \leq a_2 \leq \cdots \leq a_{2n}$ and an integer q such that $\sum_{j=1}^{2n} a_j = 2q$. Determine whether there exists a subset S of n integers such that $\sum_{i:a_i \in S} a_i = q$.

The balanced partition problem is a variant of the partition problem (in which S is not required to contain exactly n integers). The NP-completeness of the balanced partition problem is a folklore result, which follows from a slight modification on NP-completeness proof for the partition problem given by Garey and Johnson[11].

Using the balanced partition problem, we are able to show the NP-hardness of the following problem in Lemma 12.

Balanced partition′: Given a set of positive integers $\{a_1, a_2, \cdots, a_{2n}\}$ where $a_1 \leq a_2 \leq \cdots \leq a_{2n}$ and an integer q such that $\sum_{j=1}^{2n} a_j = 2q$, $a_n + a_{n+1} + \cdots + a_{2n-1} \geq q+1$. Determine whether there exists a subset S of n integers such that $\sum_{i:a_i \in S} a_i = q$.

Lemma 12. *Balanced partition′ is NP-complete.*

Proof. Membership in NP is straightforward. We show balanced partition′ is NP-hard in the following via reduction from balanced partition.

Given an instance of the balanced partition problem where the integers are $a_1 \leq a_2 \leq \cdots \leq a_{2n}$ and $q = 1/2 \cdot \sum_{j=1}^{2n} a_j$, we construct an instance of the balanced partition′ problem by adding $4n$ integers, each of value $3q$.

We first show that the constructed instance is a feasible instance. Obviously every additional integer is larger than any a_j where $j \leq 2n$. Let the additional integers be $a_{2n+1}, a_{2n+2}, \cdots, a_{6n}$. In the constructed instance there are $2n' = 6n$ elements, with the summation of all integers being $2q + 12nq$. Let $q' = q + 6nq$. Obviously $a_{n'+1} + a_{n'+2} + \cdots + a_{2n'-1} = 9nq > q' + 1$. Thus, the constructed instance is a feasible instance of the balanced partition′ problem.

We show that the constructed instance admits a feasible partition if and only if the given balanced partition instance admits a feasible solution.

If the given balanced partition instance admits a feasible solution, then obviously the constructed balanced partition′ problem admits a feasible solution (by adding $2n$ of the additional integers to both sides).

Suppose the constructed balanced partition′ instance admits a feasible solution. We claim that among the $4n$ additional integers, there are exactly $2n$ of them in S. Otherwise S contains either at most $2n - 1$ of them or at least $2n + 1$ of them. By symmetry we assume without loss of generality that S contains at most $2n - 1$ of them, then all integers in S add up to at most $(2n - 1) \cdot 3q + 2q < q' = q + 6nq$, which is a contradiction. Thus, S contains exactly $2n$ additional integers, implying that the remaining n integers adding up to q, i.e., the given balanced partition instance admits a feasible solution. □

Proof (Proof of Lemma 11). We will prove the NP-hardness for an election with only two candidates under 1-approval (plurality). Recall that in this case the constructive and destructive protection problem are the same.

We reduce from the balanced partition′ problem. Given an arbitrary instance of the balanced partition′ problem, we construct an instance of the $-bribery-protection problem such that the answer to the problem is "Yes" if and only the balanced partition′ instance admits a feasible solution.

We construct the $-bribery-protection instance as follows. There are $m = 2$ candidates. c_1 is the designated candidate. Let $F = 4qn + q$, $B = (4n - 1)q - 1$. There are $6n-1$ voters, each of unit weight and can be divided into three groups:

- $2n$ key voters voting for c_2, whose awarding prices are $4q+a_1, 4q+a_2, \cdots, 4q+a_{2n}$ and bribing prices are $4q - a_1, 4q - a_2, \cdots, 4q - a_{2n}$, respectively.
- $2n-1$ dummy voters voting for c_2, whose awarding prices are all $F + 1$ and bribing prices are all $B + 1$.
- $2n$ dummy voters voting for c_1, whose awarding and bribing prices are all 1.

Let $\mathcal{V}^* = \{v_1, v_2, \cdots, v_n\}$ be the set of key voters.

Obviously c_2 is the original winner. We show that the answer to the constructed $-bribery-protection instance is "Yes" if and only if the balanced partition′ problem admits a feasible solution.

Suppose the given balanced partition′ instance admits a feasible solution S. Now we let the defender fix the n key voters whose awarding price is $4q + a_j$ where $a_j \in S$. It is easy to verify that the total awarding price is $4nq + q = F$, which stays within the defense budget. We argue that, no briber can alter the election result with a budget of B. Let $\mathcal{V}_F$ be the set of fixed voters. Suppose on the contrary there is a briber who can make c_1 win. Given that the total attack

budget is B, the briber can only bribe key voters. Furthermore, the briber has to bribe at least n voters. As $|\mathcal{V}_F| = n$, the briber has to bribe all voters in $\mathcal{V}^* \setminus \mathcal{V}_F$. However,

$$\sum_{j:v_j \in \mathcal{V}^* \setminus \mathcal{V}_F} p_j^b = (4n-1)q > B,$$

which is a contradiction. Thus, the answer to the constructed \$-bribery-protection instance is "Yes".

Suppose the answer to the constructed \$-bribery-protection instance is "Yes". Note that in total there are $4n-1$ voters voting for c_2 and $2n$ voters voting for c_1. Thus any briber who wants to alter the election result has to bribe at least n voters who originally vote for c_2. Since the $2n-1$ dummy jobs voting for c_2 can never be protected nor bribed, we can fix a subset $\mathcal{V}_F \subseteq \mathcal{V}^*$ such that no briber can bribe n or more voters from $\mathcal{V}^* \setminus \mathcal{V}_F$ with a budget of B. We have the following claim.

Claim. $|\mathcal{V}_F| = n$.

Proof (Proof of Claim 7.5). We first show that $|\mathcal{V}_F| \leq n$. Suppose on the contrary that $|\mathcal{V}_F| \geq n+1$. Note that any key voter has an awarding price of at least $4q$, thus the total awarding price is at least $4(n+1)q$. However, $F = 4qn + q < 4(n+1)q$, which is a contradiction.

We now show that $|\mathcal{V}_F| \geq n$. Suppose on the contrary that $|\mathcal{V}_F| \leq n-1$. Then there are at least $n+1$ key voters that can be bribed and we bribe the cheapest n voters. As $p_1^b \geq p_2^b \geq \cdots \geq p_{2n}^b$, the total bribing price of the cheapest n voters among any $n+1$ voters is at most $p_n^b + p_{n+1}^b + \cdots + p_{2n-1}^b = 4qn - (a_n + a_{n+1} + \cdots + a_{2n-1}) \leq 4qn - (q+1) = B$, whereas the briber can always bribe the cheapest n voters and let c_2 win, which contradicts the fact that the answer to the \$-bribery-protection instance is "Yes". Hence $|\mathcal{V}_F| \geq n$. □

Now the following inequalities hold simultaneously:

$$\sum_{j:v_j \in \mathcal{V}_F} p_j^a \leq F = 4qn + q \tag{5a}$$

$$\sum_{j:v_i \in \mathcal{V}^* \setminus \mathcal{V}_F} p_j^b \geq B + 1 = 4(n-1/2)q + q \tag{5b}$$

Note that

$$\begin{aligned} &\textstyle\sum_{j:v_i \in \mathcal{V}^* \setminus \mathcal{V}_F} p_j^b = \sum_{j=1}^{2n} p_j^b - \sum_{j:v_j \in \mathcal{V}_F} p_j^b \\ = \quad & 4(2n-1/2)q - \textstyle\sum_{j:v_j \in \mathcal{V}_F} (4q - a_j), \end{aligned}$$

by (5a) we have

$$\sum_{j:v_j \in \mathcal{V}_F} (4q - a_j) \leq 4qn - q.$$

Using the fact that $|\mathcal{V}_F| = n$, we have

$$\sum_{j:v_j \in \mathcal{V}_F} a_j \geq q.$$

From (5b), we have

$$\sum_{j:v_j \in \mathcal{V}_F} a_j \leq q.$$

Thus,

$$\sum_{j:v_j \in \mathcal{V}_F} a_j = q,$$

i.e., the given balanced partition′ instance admits a feasible solution. □

The symmetric $-protection problem (i.e., $p_j^a = p_j^b$), however, is significantly easier, as shown by Theorem 5.

Theorem 5. *For constant m, both destructive and constructive symmetric $-protection problems are in P for any scoring rule.*

Proof. The proof idea is the same as that of Theorem 3, namely by trying all different possible $\mathcal{V}_F$. For each $\mathcal{V}_F$, we try all possible $\mathcal{V}_B$'s and all possible ways of altering the preference of voters in $\mathcal{V}_B$. Note that every voter has the same weight and satisfies that $p_j^a = p_j^b$. Therefore, a voter with a smaller (awarding and bribing) price always dominates a voter with a larger price. By Corollary 1 and Corollary 2, for the voters having the same preference, $\mathcal{V}_F$ contains the voters that have the smallest prices. Therefore, in order to determine $\mathcal{V}_F$, it suffices to know the number of voters that have the same preference, implying that there are at most $n^{m!}$ different kinds of $\mathcal{V}_F$'s. Similarly, given a $\mathcal{V}_F$, there are at most $n^{m!}$ different kinds of $\mathcal{V}_B$'s. Using the same argument as that of Theorem 3, for every $\mathcal{V}_F$ and $\mathcal{V}_B$, there are at most $n^{m!}$ ways of altering the preferences of the voters in $\mathcal{V}_B$. Therefore, there are $n^{3m!}$ different possibilities in total, which can be enumerated efficiently when m is a constant. □

7.6 Proofs Omitted in Section 4.2

The goal is to prove Theorem 7.

Theorem 7. *Both r-approval destructive weighted-protection and r-approval (symmetric) $-protection problems are NP-complete.*

Towards the proof, we first show the NP-hardness.

Lemma 13. *The r-approval destructive weighted-protection problem is NP-hard for any $r \geq 3$.*

Proof. We prove the lemma for $r = 3$. The case of $r > 3$ can be proved by introducing dummy candidates and letting each voter vote for $r - 3$ distinct dummy candidate.

We reduce from a variant of 3-dimensional matching in which every element occurs at most $d = O(1)$ times in the given triples, which is also known to be NP-hard stated by Kann [13]. Given a 3DM instance with $3\zeta = |W \cup X \cup Y|$ elements and $\eta = |M|$ triples such that every element appears at most $d = O(1)$ times in M, we construct an instance of the destructive weighted bribery-protection problem as follows. Here we further require that $\eta \geq \zeta + 2d$. The assumption is without loss of generality since if $\eta \leq \zeta + 2d - 1$, then there are at most $O(1)$ triples outside a perfect matching, and the existence of a perfect matching can be determined by brute-forcing within $\zeta^{O(1)}$ time, which is polynomial.

For ease of description, we re-index all elements of $W \cup X \cup Y$ arbitrarily as $z_1, z_2, \cdots, z_{3\zeta}$.

Let $Q = 2\eta + 1$. There are $3\zeta + 1$ key candidates, including the following two kinds of candidates (the function f will be defined later):

- 3ζ key candidates c_1 to $c_{3\zeta}$, with c_i corresponding to $z_i \in W \cup X \cup Y$ and has a score of $Q \cdot f(z_i)$. We call them element candidates;
- one key candidate $c_{3\zeta+1}$ called leading candidate, which is the original winner and has a score of $Q \cdot f(z_{3\zeta+1})$.

Besides key candidates, there are also sufficiently many dummy candidates c_i for $i > 3\zeta + 1$, each having a score of 1. The number of dummy candidates will be determined later.

There are η key voters v_1 to v_η, each of weight Q. Each key voter corresponds to a distinct triple in $(z_i, z_j, z_k) \in M$, and votes for the three candidates that correspond to z_i, z_j, z_k, respectively.

Besides key voters, there are also sufficiently many dummy voters v_j for $j > \eta$. A dummy vote has a unit weight, and votes for one key candidate and two distinct dummy candidates.

Now we determine the number of dummy voters and dummy candidates together with all the parameters. If we only consider key voters of weight Q, then every element candidate corresponding to some z_i gets a score of $Q \cdot d(z_i)$ where $d(z_i)$ is the number of occurrences of z in M. Adopting the viewpoint of the minmax vector addition problem, for every $1 \leq j \leq \eta$ we have

$$\Delta_{ij} = \begin{cases} 0, & \text{if } v_j \text{ votes for } c_i \text{ and does not vote for } c_{3\zeta+1} \\ 1, & \text{if } v_j \text{ votes for } c_{3\zeta+1} \text{ and does not votes for } c_i, \\ & \text{or } v_j \text{ votes for both } c_i \text{ and } c_{3\zeta+1} \\ 2, & \text{if } v_j \text{ votes for } c_{3\zeta+1} \text{ and does not vote for } c_i \end{cases}$$

Let

$$\Delta_{max} = \max_{1 \leq i \leq 3\zeta} \sum_{j=1}^{\zeta} \Delta_{ij}, \quad d_{max} = \max_{1 \leq i \leq 3\zeta} d(z_i).$$

We define

$$f(z_i) = 2\eta + d_{max} + \Delta_{max} - \sum_{j=1}^{\eta} \Delta_{ij}, 1 \leq i \leq 3\zeta$$

That means, candidate c_i, $1 \leq i \leq 3\zeta$ will get a score of $Q \cdot d(z_i)$ from key voters, and additionally $Q \cdot [f(z_i) - d(z_i)] \geq 0$ score from dummy voters. Hence, for each c_i we need to create $Q \cdot [f(z_i) - d(z_i)]$ dummy voters.

We define

$$f(z_{3\zeta+1}) = 2\eta + d_{max} + \Delta_{max} - \zeta + 1.$$

Note that $\sum_{j=1}^{\eta} \Delta_{ij} \geq \eta - d > \zeta$ as every element appears at most d times in triples, hence $f(z_{3\zeta+1}) > f(z_i)$ for $1 \leq i \leq 3\zeta$ and $c_{3\zeta+1}$ is indeed the original winner.

Overall, dummy voters should contribute $Q \cdot [f(z_i) - d(z_i)]$ to each c_i, $1 \leq i \leq 3\zeta$ and $(\zeta+1)f(z_{3\zeta+1})$ to $c_{3\zeta+1}$. We create in total $Q \cdot [\sum_{i=1}^{3\zeta+1} f(z_i) - \sum_{i=1}^{3\zeta} d(z_i)]$ dummy voters, and $2Q \cdot [\sum_{i=1}^{3\zeta+1} f(z_i) - \sum_{i=1}^{3\zeta} d(z_i)]$ dummy candidates.

Let the defense budget be $F = \zeta$ and the attack budget be $B = \eta - \zeta$.

"Yes" Instance of 3DM → "Yes" Instance of Destructive Weighted-Bribery-Protection. Suppose the given 3DM instance admits a feasible solution, we show that the answer to destructive weighted-bribery-protection problem is "Yes". Let $T \subseteq M$ be the perfect matching. Then $|T| = \zeta$ and we let the defender protect voters corresponding to the triples in T. Taking the viewpoint of the minmax vector addition problem. If the attacker bribes all the key voters, then $W(c_i)$ increases by exactly $Q \cdot \sum_{j=1}^{\eta} \Delta_{ij}$ for $1 \leq i \leq 3\zeta$. First it is easy to see that no dummy candidate can be a winner as $Q \cdot \sum_{j=1}^{\eta} \Delta_{ij} \leq 2\eta Q$ while $Q \cdot f(z_{3\zeta+1}) \geq (2\eta + 1)Q$. Meanwhile, for each key candidate c_i, $1 \leq i \leq 3\zeta$, his/her total score becomes exactly $Q \cdot [f(z_{3\zeta+1}) + \zeta - 1]$. As the defender fixes a subset of key voters, we should subtract the contribution of these key voters. As the triple corresponding to these voters form a perfect matching, these voters contribute a score of exactly $Q(\zeta - 1)$ to each c_i, hence after bribery every key candidate has a score at most $Q \cdot f(z_{3\zeta+1})$, implying that the answer to the Destructive Weighted-Bribery-Protection problem is "Yes".

"No" Instance of 3DM → "No" Instance of Destructive Weighted-Bribery-Protection. Suppose the given 3DM instance does not admit a perfect matching, we show that the answer to the constructed instance of the destructive weighted-bribery-protection problem is "No". Consider an arbitrary set of voters fixed by the defender and let U be the subset of key voters that are fixed. Obviously $|U| \leq \zeta$. If $|U| < \zeta$, we add arbitrary key voters into U such that its cardinality becomes ζ. Let U' be the set of these ζ key voters and let the attacker bribe the remaining $\zeta - \eta$ key voters. Again we take the viewpoint of the minmax vector addition problem. If the attacker bribes every key voter, then the total score of every key candidate c_i, $1 \leq i \leq 3\zeta$, becomes exactly $Q \cdot [f(z_{3\zeta+1}) + \zeta - 1]$. As key voters in U are not bribed, we subtract their contribution from each c_i. Note that triples corresponding to voters in U do not form a perfect matching, thus there exists some element which appears at least twice in these triples. Let

z_k be such element and we consider c_k. It is clear that $\Delta_{kj} = 0$ if v_j votes for c_k and $\Delta_{kj} = 1$ if v_j does not vote for c_k (note that key voters never vote for $c_{3\zeta+1}$). Hence $\sum_{j:v_j \in U} \Delta_{kj} \leq \zeta - 2$. By subtracting the contribution of voters in U, c_k has a score at least $Q \cdot [f(z_{3\zeta+1} + 1)]$, implying that after bribery c_k will get a higher score than $c_{3\zeta+1}$. Thus, the answer to the Destructive Weighted-Bribery-Protection problem is "No". □

Note that in the preceding reduction, we only construct voters of two different weights, Q for the key voters and 1 for the dummy voters. Recall that Q is set to be large enough to assure that only the key voters will be considered by the defender or the attacker. Once $\mathcal{V}_F$ and $\mathcal{V}_B$ are restricted to be subsets of the key voters, the concrete value of Q does not matter. Moreover, we can also prove the NP-hardness of the destructive (symmetric) $-bribery-protection problem by using essentially the same proof, except that we set key voters of price 1 and dummy voters of price exceeding budgets F and B, say, $\max\{F, B\} + 1$. This leads to the following lemma.

Lemma 14. *The r-approval destructive (symmetric) $-bribery-protection problem is NP-hard for any $r \geq 3$.*

Having showed the NP-hardness of the destructive weighted-protection problem, we show the problem is polynomial-time verifiable and is therefore NP-complete.

Lemma 15. *The destructive weighted-protection problem can be verified in polynomial time under any scoring rule.*

Proof. We leverage the *minmax vector addition* problem. In the case of unit price, given $\mathcal{V}_F$, the decision version of the verification problem becomes: does there exist a subset $\mathcal{V}_B \subseteq \mathcal{V} \setminus \mathcal{V}_F$ such that the following is true

$$\left\| \boldsymbol{\Lambda} + \sum_{j:v_j \in \mathcal{V}_B} \boldsymbol{\Delta}_j \right\| > \Lambda(c_m).$$

To answer this decision problem, it suffices to do the following for every $1 \leq i \leq m-1$: pick B voters from $\mathcal{V} \setminus \mathcal{V}_F$ whose i-th coordinate Δ_{ij} is the largest, add them to $\Lambda(c_i)$, and check if it is greater than $\Lambda(c_m)$. □

It is, however, not clear if the destructive $-protection problem is NP-complete for arbitrary scoring rules. However, we show in the following that for any scoring rule which only assigns a constant number of different scores to a preference list, i.e., the α_i's only take $O(1)$ distinct values, the $-protection problem can be verified in polynomial-time. As in the case of the r-approval rule, the α_i's only take values of 1 or 0, the destructive $-protection problem is NP-complete for the r-approval rule.

Lemma 16. *The destructive (symmetric) $-protection problem can be verified in polynomial-time in n under any scoring rule in which the α_i's only take a constant number of distinct values.*

Proof. Consider the *minmax vector addition* problem. We observe that in the case of unit weight and that the α_i's take $O(1)$ distinct values, Δ_{ij} only takes $O(1)$ distinct values. For each coordinate i, we can check if it is possible for $\Lambda(c_i)$ and Δ_{ij}'s to add up to some value strictly greater than $\Lambda(c_m)$. Note that by adding every Δ_{ij}, we need to pay a price of p_j^b, hence it is essentially the *knapsack* problem with items having arbitrary prices but only $O(1)$ distinct weights. Such a knapsack problem can be solved in polynomial-time, e.g., by simply guessing the number of items of the same weight. Among the items of the same weight, the optimal solution should take the ones with the cheapest price. □

References

1. Bredereck, R., Chen, J., Faliszewski, P., Nichterlein, A., Niedermeier, R.: Prices matter for the parameterized complexity of shift bribery. Inf. Comput. **251**, 140–164 (2016)
2. Bredereck, R., Talmon, N.: Np-hardness of two edge cover generalizations with applications to control and bribery for approval voting. Inf. Process. Lett. **116**(2), 147–152 (2016)
3. Caprara, A., Carvalho, M., Lodi, A., Woeginger, G.J.: A study on the computational complexity of the bilevel knapsack problem. SIAM J. Optim. **24**(2), 823–838 (2014)
4. Chen, J., Faliszewski, P., Niedermeier, R., Talmon, N.: Elections with few voters: Candidate control can be easy. J. Artif. Intell. Res. **60**, 937–1002 (2017)
5. Chen, L., Zhang, G.: Approximation algorithms for a bi-level knapsack problem. Theor. Comput. Sci. **497**, 1–12 (2013)
6. Dey, P., Misra, N., Narahari, Y.: Frugal bribery in voting. Theor. Comput. Sci. **676**, 15–32 (2017)
7. Erdélyi, G., Reger, C., Yang, Y.: The complexity of bribery and control in group identification. Auton. Agents Multi Agent Syst. **34**(1), 8 (2020)
8. Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L.A.: How hard is bribery in elections? J. Artif. Intell. Res. **35**, 485–532 (2009)
9. Faliszewski, P., Hemaspaandra, E., Hemaspaandra, L.A.: Weighted electoral control. J. Artif. Intell. Res. **52**, 507–542 (2015)
10. Faliszewski, P., Rothe, J.: Control and bribery in voting. In: Brandt, F., Conitzer, V., Endriss, U., Lang, J., Procaccia, A.D. (eds.) Handbook of Computational Social Choice, pp. 146–168. Cambridge University Press (2016)
11. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman (1979)
12. Kaczmarczyk, A., Faliszewski, P.: Algorithms for destructive shift bribery. Auton. Agents Multi Agent Syst. **33**(3), 275–297 (2019)
13. Kann, V.: Maximum bounded 3-dimensional matching is max snp-complete. Inf. Process. Lett. **37**(1), 27–35 (1991)
14. Knop, D., Koutecký, M., Mnich, M.: Voting and bribing in single-exponential time. In: Vollmer, H., Vallée, B. (eds.) 34th STACS, STACS 2017, March 8-11, 2017, Hannover, Germany. LIPIcs, vol. 66, pp. 46:1–46:14. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2017)
15. Lin, A.: The complexity of manipulating k-approval elections. In: Filipe, J., Fred, A.L.N. (eds.) ICAART 2011 - Proceedings of the 3rd ICAART, Volume 2 - Agents, Rome, Italy, January 28-30, 2011. pp. 212–218. SciTePress (2011)
16. McLoughlin, A.M.: The complexity of computing the covering radius of a code. IEEE Trans. Inf. Theory **30**(6), 800–804 (1984)
17. Qiu, X., Kern, W.: Improved approximation algorithms for a bilevel knapsack problem. Theor. Comput. Sci. **595**, 120–129 (2015)
18. Wang, Z., Xing, W., Fang, S.: Two-group knapsack game. Theor. Comput. Sci. **411**(7-9), 1094–1103 (2010)
19. Wrathall, C.: Complete sets and the polynomial-time hierarchy. Theor. Comput. Sci. **3**(1), 23–33 (1976)
20. Yin, Y., Vorobeychik, Y., An, B., Hazon, N.: Optimally protecting elections. In: Kambhampati, S. (ed.) Proceedings of the 25th IJCAI, IJCAI 2016, New York, NY, USA, 9-15 July 2016. pp. 538–545. IJCAI/AAAI Press (2016)