Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (IJCAI-19)

Election with Bribe-Effect Uncertainty: A Dichotomy Result

Lin Chen[1*], **Lei Xu**[2], **Shouhuai Xu**[3], **Zhimin Gao**[4] and **Weidong Shi**[5]
[1]Department of Computer Science, Texas Tech University, TX, USA
[2]Department of Computer Science, University of Texas Rio Grande Valley, TX, USA
[3]Department of Computer Science, University of Texas at San Antonio, TX, USA
[4]Department of Computer Science, Auburn University at Montgomery, AL, USA
[5]Department of Computer Science, University of Houston, TX, USA
{chenlin198662, xuleimath}@gmail.com, shxu@cs.utsa.edu, mtion@msn.com, larryshi@ymail.com

Abstract

We consider the electoral bribery problem in computational social choice. In this context, extensive studies have been carried out to analyze the computational vulnerability of various voting (or election) rules. However, essentially all prior studies assume a *deterministic* model where each voter has an associated *threshold* value, which is used as follows. A voter will take a bribe and vote according to the attacker's (i.e., briber's) preference when the amount of the bribe is above the threshold, and a voter will not take a bribe when the amount of the bribe is not above the threshold (in this case, the voter will vote according to its own preference, rather than the attacker's). In this paper, we initiate the study of a more realistic model where each voter is associated with a *willingness function*, rather than a fixed *threshold* value. The willingness function characterizes the *likelihood* a bribed voter would vote according to the attacker's preference; we call this *bribe-effect uncertainty*. We characterize the computational complexity of the electoral bribery problem in this new model. In particular, we discover a dichotomy result: a certain mathematical property of the willingness function dictates whether or not the computational hardness can serve as a deterrence to bribery attackers.

1 Introduction

Election (or voting) is a mechanism for agents in a society or multiagent system to make decisions collectively. Because of its many interesting aspects, such as algorithmic solutions and computational complexity characteristics, there is an active research field in *computational social choice* (see, for example, the book by [Brandt *et al.*, 2016] and some recent results by [Kenig and Kimelfeld, 2019; Faliszewski *et al.*, 2019; Chen *et al.*, 2019b]). One of the most fundamental problems in computational social choice is *bribery*, namely that an attacker (i.e., briber) attempts to manipulate the outcome of an election by bribing some voters to deviate from their own preferences to the attacker's preference or designated candidate. Since its introduction by [Faliszewski *et al.*, 2009], this problem has received a considerable amount of attention; see, e.g., [Brelsford *et al.*, 2008; Xia, 2012; Faliszewski *et al.*, 2015; 2011; Parkes and Xia, 2012; Faliszewski *et al.*, 2019; Chen *et al.*, 2018a; 2018b; 2019b].

Existing studies essentially make the following *binary assumption*: a voter either (i) takes a bribe that exceeds a threshold value determined by the voter and votes according to the attacker's preference, or (ii) declines a bribe that does not exceed the threshold value and votes according to the voter's own preference. This binary assumption oversimplifies the problem because in the real world a voter's decision may depend on the amount of the bribe. For example, a voter, who takes a bribe worth $101 because its threshold value is $100, may also take a bribe worth $99 with some probability.

The aforementioned inadequacy of the binary assumption has actually been discussed by researchers in the fields of psychology (see, e.g., [Gerlach *et al.*, 2019]) and economic behavior (see, e.g., [Frank and Schulze, 2000]). However, the computational social choice community does not appear to be aware of these studies until now.

In this paper, we cope with the inadequacy of the binary assumption made in the literature of computational social choice, by investigating the following notion of *bribe-effect uncertainty*: Each voter is associated with a *willingness function*, rather than a threshold value; the willingness function determines the probability that a bribed voter will indeed vote according to the attacker's preference, where the probability varies with the amount of the bribe. With this new perspective, the classic bribery problem becomes what we call the *Election with Bribe-Effect Uncertainty* (EBEU) problem. The decision version of the EBEU problem is: Can an attacker with a fixed bribery budget succeed in manipulating the outcome of an election with a probability exceeding a given threshold of interest? Correspondingly, the optimization version of the EBEU problem asks for a solution that maximizes such a probability.

Obtaining the willingness functions as an input data can be achieved via experiments. Indeed, researchers in the psychology and economic behavior community have conducted experiments to study the relationship between dishonest behavior and reward magnitude; see Section 1.2. In particular, [Gerlach *et al.*, 2019] provides a comprehensive survey over

*Contact Author

hundreds of such experiments.

It is worth stressing that a bribed voter may *not* always have to vote according to the attacker's preference, especially in elections using secret ballots. On the other hand, the EBEU model actually can be equally applied to describe the following lobbying problem: An attacker (i.e., briber) may donate an amount of money to (for example) a politician, in hoping that the politician will vote according to the attacker's preference (e.g., in deciding some public policy). In this case, the politician would certainly accept the donation but may not vote according to this particular donner's preference; instead, the politician may vote according to the preference of another donner who donates possibly a bigger amount of money. That is, the decision of the politician would be a function of the amount of the donation.

1.1 Our Contributions

The conceptual contribution of the paper is the introduction of a new type of uncertainty, namely bribe-effect uncertainty, into election models. This uncertainty is described by a willingness function, which eliminates the aforementioned over-simplifying binary assumption that has been widely made in the literature.

The technical contribution of the paper is the following dichotomy result. On one hand, we show that the EBEU problem under the *Plurality* voting rule, which will be explained later, does *not* admit any $O(1)$-approximation FPT-algorithm for *arbitrary* willingness functions, assuming FPT$\neq$W[1]. This means that the computational complexity of the EBEU problem could serve as an effective deterrence to bribery attackers. On the other hand, we show that if the logarithm of *every* voter's willingness function is Lipschitz continuous (which will be defined later), then there exists an FPT algorithm that produces $(1+\varepsilon)$-approximate solution, meaning that the computational complexity may not be able to deter bribery attackers when their willingness functions are "smooth". This result is interesting because it's the mathematical property, rather than any specific form, of the willingness function that dictates whether or not the computational hardness can serve as a deterrence.

1.2 Related Work

Uncertainty is inherent to many real-world complex problems and coping with it has become a fundamental research problem. Putting into the context of election, uncertainty is inherent to the bribery problem because it is inadequate to model voters as machines that return "yes" or "no" based on whether or not a monetary award exceeds a threshold (see, e.g., [Frank and Schulze, 2000]). Nevertheless, the *bribe-effect uncertainty* we study in the present paper has not been investigated in the literature. The most closely-related prior works are [Chen *et al.*, 2019b] and [Wojtas and Faliszewski, 2012], which still make the *binary assumption* despite that some bribed voters have some "no-show" probabilities (i.e., they may not vote at all). In our model, we eliminate the binary assumption and replace it with willingness functions, which characterize the relationship between the amount of bribe and the probability that a bribed voter indeed votes according to the attacker's preference. This distinction between the different kinds of uncertainty is also confirmed by the fact that the problem studied by [Chen *et al.*, 2019b] always denies an $O(1)$-approximation FPT algorithm, while the approximability of our problem is crucially dependent upon a certain mathematical property of the willingness functions, as indicated by the dichotomy result mentioned above.

It is worth mentioning that other kinds of uncertainty (e.g., margin of victory), which are loosely related to the bribe-effect uncertainty, have been studied by [Dey and Narahari, 2015; Erdelyi *et al.*, 2014; Mattei *et al.*, 2015]. It is also worth mentioning that we focus on investigating the impact of the mathematical property of the willingness function, rather than its specific form, which is actually an open problem. Indeed, some researchers argue that a larger "reward" (or bribe) would increase the chance of dishonest behavior (see [Conrads *et al.*, 2014; Gneezy, 2005]); others actually argue for the opposite — a larger bribe may lead to a smaller chance of dishonest behavior — because the psychological cost of cheating may increase (see [Mazar *et al.*, 2008]); yet others argue that they are relatively independent (see [Abeler *et al.*, 2016]). Our dichotomy result applies regardless of the correctness of these arguments because we show that it is *not* the monotonicity of the willingness function that matters most in determing whether or not the computational complexity can serve as a deterrence.

Despite the classical bribery model that assumes a threshold bribery cost for each voter [Faliszewski *et al.*, 2009], other kinds of bribery model like swap bribery are also considered, see, e.g. [Elkind *et al.*, 2009; Bredereck *et al.*, 2016a; 2016b; Elkind and Faliszewski, 2010].

2 Problem Statement and Preliminaries

Election Problem. There are m candidates, denoted by a set $\mathscr{C}=\{c_1,c_2,\ldots,c_m\}$, and n voters, denoted by a set $\mathscr{V}=\{v_1,v_2,\ldots,v_n\}$. Each voter votes according to its preference over the candidates $c_1,c_2,\ldots,c_m$. There is a voting rule, according to which a winner is determined. There are many voting rules, but we focus on the *plurality rule*, which says that each voter votes for its most preferred candidate and the candidate receiving the most votes will be the winner.

Bribery Problem. In this problem, an attacker (i.e., briber) attempts to manipulate the outcome of an election by bribing some voters that would deviate from voting for their own preferred candidate to voting for the attacker's designated candidate. Specifically, let voter v_i has a bribery price q_i, meaning that receiving a bribe worth q_i will make v_i vote for the attacker's designated candidate, regardless of v_i's own preference. The attacker has a total budget Q that can be spent on bribing voters.

EBEU Problem. This problem extends the bibery problem, which uses a binary willingness function $f_j:\mathbb{R}_{\geq 0}\to\{0,1\}$, with a more general willingness function $f_j:\mathbb{R}_{\geq 0}\to[0,1]$ for voter v_j such that $f_j(x)$ returns the probability that v_j will vote for the attacker's designated candidate, where x is the amount of bribe received from the attacker and $1\leq j\leq n$. Without loss of generality, let c_1 be the winner when there are no bribery attacks and c_m be the attacker's designated candidate. Suppose the attacker has a fixed budget Q for waging the bribery attack and each voter v_j has a willingness functon f_j. The EBEU problem asks for identifying a subset of k voters in $V' \subseteq V$, each of which receives a bribe of amount x_j where $v_j \in V'$, such that the probability that the attacker's designated candidate c_m wins the election (i.e., the attacker succeeds in manipulating the election) is maximized.

Formally, the EBEU problem is described as follows while normalizing the attacker's budget Q to 1 for a technical convenience.

The (Plurality-)EBEU Problem
Input: A set of m candidates $\mathscr{C} = \{c_1, c_2, \ldots, c_m\}$, where c_1 is the winner in the absence of bribery attacks and c_m is the attacker's designated candidate; a set of n voters $\mathscr{V} = \{v_1, v_2, \cdots, v_n\}$; a positive integer k; an attack budget (normalized to) 1; each voter $v_j \in \mathscr{V}$ is associated with a willingness function f_j such that if v_j receives a bribe of amount x from the attacker, then v_j will vote, with probability $f_j(x)$, according to the attacker's preference rather than v_j's own preference (in the case of the plurality voting rule, v_j will vote for the attacker's designated candidate c_m).
Output: Find a set of indices $I^* \subseteq \{1, 2, \cdots, n\}$, $|I^*| = k$, together with $x_j \in \mathbb{R}_{\geq 0}$ for each $j \in I^*$ such that

- $\sum_{j \in I^*} x_j \leq 1$, and
- the probability that c_m wins the election (under the plurality voting rule) is maximized by bribing voters belonging to $V^* = \{v_i \in V \setminus V_m | i \in I^*\}$.

Lipschitz Continuity. Since we will show that the *Lipschitz continuity* of the willingness function $f_j(\cdot)$ will play the critical role in determing whether the election problem under bribe-taking uncertainty is vulnerable to the bribery attack or not, we need to review this property.

Definition 1 (Lipschitz continuity)**.** *Given two metric space (X, d_X) and (Y, d_Y), where d_X and d_Y respectively denote the metrics in X and Y. A function $f : X \to Y$ is said* Lipschitz continuous *if there exists a universal real constant $\alpha_0 \geq 0$ such that for all $x_1, x_2 \in X$, it holds that*

$$d_Y(f(x_1), f(x_2)) \leq \alpha_0 \cdot d_X(x_1, x_2). \quad (1)$$

When the function f is defined on real numbers, which is true in the setting of the present paper, the condition specified by Eq. (1) *can be rewritten as*

$$|f(x_1) - f(x_2)| \leq \alpha_0 \cdot |x_1 - x_2|. \quad (2)$$

3 Hardness of EBEU With Non-"Lipschitz Continuous" Willingness

In this section, we show via Theorem 1 that if some of the $\log f_j(\cdot)$'s are *not* Lipschitz continuous, then the EBEU problem does not admit any constant ratio approximation algorithms. The inapproximability holds even if the willingness functions are continuous. The implication of this hardness result is that election under bribe-effect uncertainty is *not* vulnerable to *optimal* bribery attacks, namely that the complexity in finding an optimal attack may hinder the attacker from waging such attacks.

Theorem 1 (Main hardness result)**.** *Assuming $W[1] \neq FPT$, there exist (continuous) willingness functions, $f_j(\cdot)$'s, such that the EBEU problem does not admit any $g(k)$-approximation algorithm that runs in FPT time parameterized by k for any computable function g, even if $m = 2$.*

In order to prove Theorem 1, we leverage the 2-dimensional knapsack problem, which is reviewed below, and its W[1]-hardness result owing to [Kulik and Shachnai, 2010].

The 2-dimensional Knapsack
Input: A set of n' items, where each item j has a 2-dimensional size $(a_j, b_j) \in \mathbb{Z}^2_{\geq 0}$; a 2-dimensional knapsack of size $(A, B) \in \mathbb{Z}^2_{>0}$.
Output: Decide whether or not there exists a subset S of items such that $|S| = r$ and $\sum_{j \in S}(a_j, b_j) \leq (A, B)$.

Theorem 2 (Theorem 7, [Kulik and Shachnai, 2010])**.** *Assuming $W[1] \neq FPT$, there does* not *exist any algorithm that runs in time $f_{KP}(r)|I_{KP}|^{O(1)}$ for solving the 2-dimensional knapsack problem for any computable function f_{KP}, where $|I_{KP}|$ is the length of the input.*

The strategy for proving Theorem 1 is the following: Suppose on the contrary that there exists some α-approximation FPT algorithm that solves the EBEU problem in $f_{EBEU}(k)|I_{EBEU}|^{O(1)}$ time for some computable function f_{EBEU}, where $\alpha = g(k)$ for some function g, we can show that this algorithm can be utilized to solve the 2-dimensional knapsack problem in $f_{KP}(r)|I|^{O(1)}$ time for some computable function f_{KP}. This contradicts with Theorem 2.

Proof of Theorem 1. Under the proof strategy mentioned above, we first construct an instance of the EBEU problem from an instance of the 2-dimensional knapsack problem according to the following two steps. First, we construct two candidates c_1 and c_2, where c_1 is the winner when there are no bribery attacks and c_2 is the attacker's designated candidate. Recall that the bribe budget is defined to be 1. Second, we construct $n = 2n' + 2k - 1$ voters, including n' *key* voters, each of which corresponds to an item, and $n' + 2k - 1$ *dummy* voters, each of which does not correspond to any item, where $k = r$. The difference between these two types of voters is in their willingness functions: the willingness functions of *key* voters are *not* Lipschitz continuous, but the willingness functions of *dummy* voters are Lipschitz continuous.

- Constructing key voters: For each item j of 2-dimensional size (a_j, b_j), a key voter v_j is constructed with the following willingness function f_j:

$$f_j(x) = \begin{cases} 0, & \text{if } x < \frac{A+a_j-\delta}{(k+1)A} \\ \frac{x-A-a_j+\delta}{\delta} M^{-b_j} & \text{if } \frac{A+a_j-\delta}{(k+1)A} \leq x \leq \frac{A+a_j}{(k+1)A} \\ M^{-b_j}, & \text{if } \frac{A+a_j}{(k+1)A} < x \leq 1 \\ x - 1 + M^{-b_j}, & \text{if} 1 < x \leq 2 - M^{-b_j} \\ 1, & \text{otherwise} \end{cases}$$

  where $M > \alpha$ is an integer (e.g., $M = \alpha + 1$) and δ is a sufficiently small rational number (e.g., $\delta = 1/(100n)$).

Note that the function f_j is *continuous*, but it has a sharp increase around the point $\frac{A+a_j}{(k+1)A}$, where the value explodes with rate $O(1/\delta) = O(n)$. Hence, f_j and $\log f_j$ are *not* Lipschitz continuous.

- Constructing dummy voters: Each dummy voter has the following willingness function f_{dummy}:

$$f_{dummy}(x) = \begin{cases} 0, & \text{if } x \leq 1 \\ x-1, & \text{if } 1 \leq x \leq 2 \\ 1, & \text{otherwise.} \end{cases}$$

All the n' key voters vote for c_1 because they are not bribed by the attacker. Among the $n'+2k-1$ dummy voters, $2k-1$ of them vote for c_1 and n' of them vote for c_2. This completes the construction of a EBEU instance.

Now, suppose there exists a α-approximation FPT algorithm for the EBEU problem that runs in $f_{EBEU}(k)|I_{EBEU}|^{O(1)}$ time. Then, we can use this algorithm to solve the given 2-dimensional knapsack instance, yielding a contradiction. Recall that $k = r$ and that there is a one-to-one correspondence between the key voters in the constructed EBEU instance and the items in the given 2-dimensional knapsack instance. The proof is based on the following 5 claims (see [Chen *et al.*, 2019a] for the omitted proofs).

Claim 1. *If c_2 wins with probability 0 in the approximation solution to the constructed EBEU instance, the given 2-dimensional knapsack instance does not admit any feasible solution.*

From now on we assume c_2 wins with a positive probability in the approximation solution to the EBEU instance. Note that if the attacker chooses to bribe some voter, the attacker should spend an amount such that the voter will vote for the attacker's designated candidate with a positive probability. This means that if the attacker chooses to bribe a dummy voter, the attacker should spend an amount that is strictly larger than 1, which is impossible. Hence, the attacker bribes exactly k key voters in any feasible solution. Let V' be an arbitrary feasible solution to the EBEU instance, and let S' be the corresponding subset of items in the 2-dimensional knapsack instance. It is clear that $j \in S'$ and $v_j \in V'$ are equivalent.

Claim 2. $\sum_{j:v_j \in V'} a_j \leq A$.

Claim 3. *Without loss of generality, we can assume that the attacker bribes $v_j \in V'$ with an amount $\frac{A+a_j}{(k+1)A}$.*

Claim 4. *If the given 2-dimensional knapsack instance admits a feasible solution, then the objective value of an α-approximation solution to the EBEU instance is at least M^{-B}.*

Claim 5. *If the given 2-dimensional knapsack instance does not admit any feasible solution, then the objective value of the approximation solution to the EBEU instance is at most M^{-B-1}.*

By Claim 1, Claim 4 and Claim 5, we can decide whether the given 2-dimensional knapsack instance admits a feasible solution by checking whether or not the α-approximation solution to the EBEU instance has an objective value larger than M^{-B-1}. Since the approximation algorithm runs in $f_{EBEU}(k)|I_{EBEU}|^{O(1)}$ time for some computable function f_{EBEU} and that $r = k$, we derive an FPT algorithm for the 2-dimensional knapsack problem, contradicting Theorem 2. Hence, Theorem 1 holds. □

4 FPT-Approximation Schemes for EBEU With Lipschitz Continuous Willingness

Now we present an algorithmic result in Theorem 3, while assuming the willingness functions are Lipschitz continuous.

Theorem 3 (Main algorithmic result). *Let $F_j^+ = \{x : f_j(x) > 0\}$ where $1 \leq j \leq n$. If $\log f_j(x)$ is Lipschitz continous for all $x \in F_j^+$ as well as $1 \leq j \leq n$ and the number of candidates m is a constant, then there exists an algorithm for solving the EBEU problem such that the algorithm runs in $f_{EBEU}(k)|I_{EBEU}|^{O(1)}$ time for some computable function f_{EBEU} and returns a solution with an objective value that is no smaller than $(1-\varepsilon)$OPT, where OPT $\in [0,1]$ is the optimal objective value and $\varepsilon > 0$ is an arbitrary small constant.*

In order to prove Theorem 3, we proceed as follows. In Section 4.1, we show the existence of a *well-structured near optimal solution*. In Section 4.2, we show how to guess important structural information for identifying the *well-structured near optimal solution*. In Section 4.3, we present an approximation algorithm that returns a $k^{O(k)}$-approximation solution. This approximation algorithm provides an upper bound of the optimal objective value, through which we develop a dynamic programming-based FPT approximation scheme in Section 4.4.

4.1 Existence of a Near Optimal Solution

Recall that the total budget is 1 and we only consider $f_j(x)$ where $x \leq 1$. The following property of $f_j(\cdot)$'s plays a crucial role in deriving a $k^{O(k)}$-approximation algorithms, which leads to an FPT approximation scheme. Intuitive, $\ln f_j$ being Lipschitz continuity means that the value of $f_j(x)$ does not increase arbitrary as x increases, as is shown by Corollary 1. This fact is particularly useful in two aspects. First, we can round down the cost spent on bribing each voter by some sufficiently small amount without causing the value of the willingness function to change much. This allows us to show the existence of a well-structured near optimal solution. Second, we can derive a $k^{O(k)}$-approximation solution through the following heuristic: If we have a budget of amount k instead of 1, then we can simply bribe the k voters whose $f_j(1)$'s are the largest; given that we only have a budget of 1, we can choose to spend $1/k$ to bribe each of these voters, and this greedy solution would not be too far from the optimal one because $f_j(1)$ and $f_j(1/k)$ do not differ too much, owing to the property of Lipschitz continuity.

The following Lemma 1, Lemma 2 and Corollary 1 are all deduced from Lipschitz continuity (see [Chen *et al.*, 2019a] for their proofs).

Lemma 1. *If* $\ln f_j(x)$ *is Lipschitz continuous for* $x \in F_j^+ \cap [0,1]$*, then*

$$|f_j((1 \pm \varepsilon)x) - f(x)| \le O(\varepsilon)f(x)$$

holds for any sufficiently small $\varepsilon > 0$.

Note that we do not necessarily restrict f_j's to be non-decreasing, but if $f_j(x) < f_j(y)$ for some $x > y$ and the attacker allocates a budget of amount x to bribe v_j, then the attacker may simply choose to spend a smaller amount to bribe v_j. For example, the attacker can spend an amount x' to bribe v_j, where $f_j(x') = \sup_{t \le x} f_j(t)$. Consequently, we define $\bar{f}_j$ as:

$$\phi_j(x) = \sup_{t \le x} f_j(t).$$

Similar to Lemma 1, the following lemma holds for function ϕ_j.

Lemma 2. *If* $\ln f_j(x)$ *is Lipschitz continuous for* $x \in F_j^+ \cap [0,1]$*, then*

$$|\phi_j((1 \pm \varepsilon)x) - \phi_j(x)| \le O(\varepsilon)\phi_j(x)$$

holds for any sufficiently small $\varepsilon > 0$.

From now on we only need to focus on $\phi_j(x)$ instead of $f_j(x)$ because the monotonicity of $\phi_j(x)$ makes our presentation easier to follow. According to Lemma 2, we have the following corollary.

Corollary 1. *If* $\ln f_j(x)$ *is Lipschitz continuous for* $x \in F_j^+ \cap [0,1]$*, then*

$$\max\{\frac{\phi_j(y)}{\phi_j(x)}, \frac{\phi_j(x)}{\phi_j(y)}\} \le (\frac{y}{x})^{O(1)}$$

holds for any $x, y \in F_j^+ \cap [0,1], x < y$.

Consider an arbitrary solution where the attacker bribes some subset V' of voters such that any $v_j \in V'$ will vote for the attacker's designated candidate with some probability p_j. Let π_1 be the probability that c_m wins. Let $v_{j_0} \in V'$ be an arbitrary fixed voter. Suppose we change the probability associated to $v_{j_0} \in V'$ from p_{j_0} to $p'_{j_0} \ge p_{j_0}$, and let π_2 be the probability that c_m wins as a consequence of the change in probability. Since v_{j_0} votes for the attacker's designated candidate with a higher probability now, it is straightforward to see that $\pi_2 \ge \pi_1$. Lemma 3 below says that π_2 cannot be too large.

Lemma 3. $\pi_2 \le \pi_1 \cdot \frac{p'_{j_0}}{p_{j_0}}$.

Proof. Let Ω be the event that when v_{j_0} votes for the attacker's designated candidate c_m and c_m wins. Let Ω' be the event that when v_{j_0} does not vote for the attacker's designated candidate c_m and c_m wins. Then, we have

$$\pi_1 = \Pr(\Omega)p_{j_0} + \Pr(\Omega')(1 - p_{j_0}),$$

and

$$\pi_2 = \Pr(\Omega)p'_{j_0} + \Pr(\Omega')(1 - p'_{j_0}).$$

Since $p_{j_0} \le p'_{j_0}$, we have $\Pr(\Omega')(1 - p_{j_0}) \ge \Pr(\Omega')(1 - p'_{j_0})$. Hence, we have $\pi_2 \le \pi_1 \cdot \frac{p'_{j_0}}{p_{j_0}}$. □

From Lemma 3, we obtain the following corollary.

Corollary 2. *Let* π' *be the probability that* c_m *wins when we change the probability that* v_j *votes for the attacker's designated candidate from* p_j *to* p'_j*. Then, we have*

$$\pi' \le \pi_1 \prod_{j \in V'} \max\{1, \frac{p'_j}{p_j}\}.$$

Note that we can interpret Lemma 3 as if we decrease the probability of v_{j_0} from p'_{j_0} to p_{j_0}, in which case the probability that v_j votes for the attacker's designated candidate decreases from π_2 to π_1, but we can still obtain the lower bound π_1 such that $\pi_1 \ge \pi_2 \cdot \frac{p_{j_0}}{p'_{j_0}}$. This leads to the following corollary:

Corollary 3. *Let* π' *be the probability that* c_m *wins when we change the probability that* v_j *votes for the attacker's designated candidate from* p_j *to* p'_j*, then we have*

$$\pi' \ge \pi_1 \prod_{j \in V'} \min\left\{1, \frac{p'_j}{p_j}\right\}.$$

Now we are ready to construct a solution. From now on we denote by V^* the subset of voters selected by the optimal solution. Let x_j be the amount of budget the attacker spends on bribing voter $v_j \in V^*$, $\phi_j(x_j) = p_j$, and π^* be the probability that c_m wins. We modify the optimal solution in the following three steps.

Step 1. We reduce the amount of budget that is spent on each voter by a factor of $1 - \varepsilon/k$, meaning that the attacker spends $(1 - \varepsilon/k)x_j$ to bribe voter $v_j \in V^*$.

Lemma 4. *After Step 1,* c_m *wins with a probability at least* $\pi^*(1 - O(\varepsilon))$.

Proof. According to Lemma 2, we have $\phi_j((1 - \varepsilon/k)x_j) \ge (1 - O(\varepsilon/k))p_j$. According to Corollary 3, the probability that c_m wins after the modification specified in Step 1 is at least $\pi^*(1 - O(\varepsilon/k))^k \ge \pi^*(1 - O(\varepsilon))$. □

Step 2. Note that after Step 1, the total amount of budget spent by the attacker is at most $1 - \varepsilon/k$. If the attacker spends less than ε/k^2 on some voter, then we increase the amount to be ε/k^2. Since at most k voters are selected, the overall increase in the spent budget is ε/k, which is still legitimate (i.e., no greater than the original total budget of 1). Note that by doing so the probability that c_m wins does not decrease and is at least $\pi^*(1 - O(\varepsilon))$.

Step 3. Consider the budget spent to bribe $v_j \in V^*$ after Step 2. We round down this amount to the nearest value in the form of $\varepsilon/k^2(1 + \varepsilon/k)^i$ for some integer $i \ge 0$. Note that this step is similar to Step 1 and using the same argument as in Step 1, we can show that after Step 3 c_m wins with a probability at least $\pi^*(1 - O(\varepsilon))$.

After conducting the preceding three steps, we call the resulting solution a *well-structured feasible solution*, which has a near optimal objective value (i.e., *well-structured near optimal solution*).

4.2 Enumeration

In order to find a *well-structured near optimal solution*, we need to guess (through enumeration) on some component in this solution. Since the amount of budget spent on each selected voter is in the form of $\varepsilon/k^2(1+\varepsilon/k)^h$ where $h \le O(k/\varepsilon \cdot \log(k/\varepsilon))$, there are only $O(k/\varepsilon \cdot \log(k/\varepsilon))$ possibilities. We now classify the voters into m groups, where V_i is the set of voters who vote for candidate c_i when there are no bribery attacks. We first guess, via k^m enumerations, the number of voters bribed in each V_i. Suppose k_i voters that belong to V_i are bribed.

For each bribed voters in V_i, the attacker spends a budget of amount $\varepsilon/k^2(1+\varepsilon/k)^h$ to bribe the voter. We can list the k_i different amounts the attacker spent to bribe the voters in V_i as a vector, leading to a k_i-dimensional vector where each element (or coordinate) can take at most $O(k/\varepsilon \cdot \log(k/\varepsilon))$ different values. We call such a vector a *package* for V_i. Through $O(k^k/\varepsilon^k \cdot \log^k(k/\varepsilon))$ enumerations, we can guess the package for each V_i. Hence, by $O(k^{mk}/\varepsilon^{mk} \cdot \log^{mk}(k/\varepsilon))$ enumerations, we can guess all of the packages.

Suppose the package for V_i is (a,b). Then, what remains to be done is to decide to select which of the two voters in V_i. Note that even if we know the two selected voters are v_{j_1} and v_{j_2}, it is far from clear that the attacker should spend budget a to bribe voter v_{j_1} and budget b to bribe voter v_{j_2}, or the attacker should spend b to bribe v_{j_1} and a to bribe v_{j_2}. In order to resolve this issue, we employ a dynamic programming approach. For this purpose, we need a $g(k)$-approximation algorithms that can provide us with a reasonable lower bound on the optimal objective value. Section 4.3 presents such an approximation algorithm.

4.3 A Simple Approximation Algorithm

Theorem 4. *If $\ln f_j(x)$ is Lipschitz continuous for $x \in F_j^* \cap [0,1]$ and $1 \le j \le n$, then there exists a $k^{O(k)}$-approximation algorithm that runs in $O(k^m|I_{EBEU}|)$ time for solving the EBEU problem, where $|I_{EBEU}|$ is the length of the input.*

In order to prove Theorem 4, we first show a general result on comparing two arbitrary solutions. Let V^1 and V^2 denote the subsets of k voters selected by two feasible solutions Sol_1 and Sol_2, respectively. Let $V_i^h = V_i \cap V^h$ for $h = 1,2$. We say "the second solution is λ-*bounded* by the first solution" if (i) $|V_i^1| = |V_i^2|$ for every i and (ii) there exists a one-to-one λ-*mapping*, denoted by σ, from the voters in V_i^1 to the voters in V_i^2, where a mapping $\sigma: V_i^1 \to V_i^2$ is called λ-*mapping* if for any $j \in V_i^1$, we have

$$\phi_{\sigma(j)}(x'_{\sigma(j)}) \le \lambda \phi_j(x_j),$$

where x_j is the amount of money the attacker spends to bribe voter v_j in the first solution, and $x'_{\sigma(j)}$ is the amount of money the attacker spends to bribe voter $v_{\sigma(j)}$ in the second solution.

Lemma 5. *Given two feasible solutions Sol_1 and Sol_2. Let π_1 and π_2 be their optimal objective values, respectively. If the second solution is λ-bounded by the first solution for some $\lambda \ge 1$, then we have $\pi_2 \le \lambda^k \pi_1$.*

Note that Theorem 4 already contrasts sharply with Theorem 1 and the running time is polynomial when m is constant.

4.4 An Approximation Scheme in FPT-Time

Theorem 5. *If $\ln f_j(x)$ is Lipschitz continuous for $x \in F_j^* \cap [0,1]$ and $1 \le j \le n$, and m is a constant, then there exists a $(1+\varepsilon)$-approximation algorithm that runs in FPT-time parameterized by k for solving the EBEU problem, where $\varepsilon > 0$ is any constant.*

Proof sketch. We will use dynamic programming to keep track of all possible "partial solutions" and find out the well-structured near optimal solution. A partial solution is a subset of voters selected among the first γ voters. A *state* for a partial solution contains the following information: i). The number of bribed voters in each V_i. ii). The total amount of cost spent so far on the bribed voters. Notice that while the cost spent on each bribed voter can be an arbitrary real number, our rounding procedure ensures that we may assume the cost to take at most $O(k/\varepsilon \cdot \log(k/\varepsilon))$ distinct values (see Section 4.2), therefore the cost spent on each bribed voter can be stored, and consequently the overall cost. iii). The random variable that corresponds to the bribed voters in each V_i in the partial solution. Note that for each bribed voter, whether or not the voter changes preference is a binary random variable, and the eventual number of votes received by each candidate after bribery depends on the summation of these m random variables. Each of these random variables can only take values in $\{0,1,\cdots,k\}$, therefore, it suffices to store its probability on each value. The probability can be arbitrary real number, so we need to round it. Rounding is possible because the approximation algorithm derived in Section 4.3 can be used to provide us with an upper and lower bounds on the optimal probability that differ by a factor of at most $k^{O(k)}$. Therefore, we can round the probabilities into an FPT number of distinct values based on the lower bound and show that the rounding will not cause the objective to increase by an $O(\varepsilon)$ factor. □

5 Conclusion and Discussion

We introduced a new perspective of the electoral bribery problem in *bribe-effect uncertainty*, which goes beyond previous studies that assume a fixed threshold value according to which a voter decides to accept or decline a bribe. We used the notion of *willingness function* to accommodate the bribe-effect uncertainty. We proved a dichotomy result, which shows that the Lipschitz continuity of the logarithm of the willingness function, rather than any specific form of the willingness function, dictates whether or not the computational hardness can serve as a deterrence to bribery attackers. The new perspective of bribe-effect uncertainty indicates that there are many interesting problems for future research. For example, we only investigated the Plurality voting rule; it is interesting to know whether the dichotomy result is applicable to other voting rules or not.

Acknowledgements

We thank Dr. Paul Romanowich for useful discussions at an early stage of this research. This work is supported in part by NSF Grant #1756014 and ARO Grant #W911NF-17-1-0566.

References

[Abeler *et al.*, 2016] Johannes Abeler, Daniele Nosenzo, and Collin Raymond. Preferences for truth-telling. 2016.

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

[Bredereck *et al.*, 2016a] Robert Bredereck, Jiehua Chen, Piotr Faliszewski, André Nichterlein, and Rolf Niedermeier. Prices matter for the parameterized complexity of shift bribery. *Information and Computation*, 251:140–164, 2016.

[Bredereck *et al.*, 2016b] Robert Bredereck, Piotr Faliszewski, Rolf Niedermeier, and Nimrod Talmon. Complexity of shift bribery in committee elections. In *AAAI*, pages 2452–2458, 2016.

[Brelsford *et al.*, 2008] Eric Brelsford, Piotr Faliszewski, Edith Hemaspaandra, Henning Schnoor, and Ilka Schnoor. Approximability of manipulating elections. In *AAAI*, pages 44–49, 2008.

[Chen *et al.*, 2018a] Lin Chen, Lei Xu, Zhimin Gao, Nolan Shah, Ton Chanh Le, Yang Lu, and Weidong Shi. The game among bribers in a smart contract system. In *Financial Cryptography and Data Security*, pages 294–307, 2018.

[Chen *et al.*, 2018b] Lin Chen, Lei Xu, Shouhuai Xu, Zhimin Gao, Nolan Shah, Yang Lu, and Weidong Shi. Protecting election from bribery: new approach and computational complexity characterization. In *AAMAS*, pages 1894–1896. IFAAMAS, 2018.

[Chen *et al.*, 2019a] Lin Chen, Lei Xu, Shouhuai Xu, Zhimin Gao, and Weidong Shi. Election with bribe-effect uncertainty: A dichotomy result. *https://github.com/IJCAIpaper/willingness/blob/master/Full-version.pdf*, 2019.

[Chen *et al.*, 2019b] Lin Chen, Lei Xu, Shouhuai Xu, Zhimin Gao, and Weidong Shi. Election with bribed voter uncertainty: Hardness and approximation algorithm. In *AAAI*, 2019.

[Conrads *et al.*, 2014] Julian Conrads, Bernd Irlenbusch, Rainer Michael Rilke, Anne Schielke, and Gari Walkowitz. Honesty in tournaments. *Economics Letters*, 123(1):90–93, 2014.

[Dey and Narahari, 2015] Palash Dey and Y Narahari. Estimating the margin of victory of an election using sampling. In *Twenty-Fourth International Joint Conference on Artificial Intelligence*, 2015.

[Elkind and Faliszewski, 2010] Edith Elkind and Piotr Faliszewski. Approximation algorithms for campaign management. In *Internet and Network Economics*, pages 473–482, 2010.

[Elkind *et al.*, 2009] Edith Elkind, Piotr Faliszewski, and Arkadii Slinko. Swap bribery. In *International Symposium on Algorithmic Game Theory*, pages 299–310. Springer, 2009.

[Erdelyi *et al.*, 2014] Gabor Erdelyi, Edith Hemaspaandra, and Lane A Hemaspaandra. Bribery and voter control under voting-rule uncertainty. In *AAMAS*, pages 61–68, 2014.

[Faliszewski *et al.*, 2009] Piotr Faliszewski, Edith Hemaspaandra, and Lane A Hemaspaandra. How hard is bribery in elections? *Journal of Artificial Intelligence Research*, 35:485–532, 2009.

[Faliszewski *et al.*, 2011] Piotr Faliszewski, Edith Hemaspaandra, and Lane A Hemaspaandra. Multimode control attacks on elections. *Journal of Artificial Intelligence Research*, 40:305–351, 2011.

[Faliszewski *et al.*, 2015] Piotr Faliszewski, Yannick Reisch, Jörg Rothe, and Lena Schend. Complexity of manipulation, bribery, and campaign management in bucklin and fallback voting. *AAMAS*, 29(6):1091–1124, 2015.

[Faliszewski *et al.*, 2019] Piotr Faliszewski, Pasin Manurangsi, and Krzysztof Sornat. Approximation and hardness of shift-bribery. In *AAAI*, 2019.

[Frank and Schulze, 2000] Björn Frank and Günther G Schulze. Does economics make citizens corrupt? *Journal of economic behavior & organization*, 43(1):101–113, 2000.

[Gerlach *et al.*, 2019] Philipp Gerlach, Kinneret Teodorescu, and Ralph Hertwig. The truth about lies: A meta-analysis on dishonest behavior. *Psychological bulletin*, 145(1):1, 2019.

[Gneezy, 2005] Uri Gneezy. Deception: The role of consequences. *American Economic Review*, 95(1):384–394, 2005.

[Kenig and Kimelfeld, 2019] Batya Kenig and Benny Kimelfeld. Approximate inference of outcomes in probabilistic elections. In *AAAI*, 2019.

[Kulik and Shachnai, 2010] Ariel Kulik and Hadas Shachnai. There is no eptas for two-dimensional knapsack. *Information Processing Letters*, 110(16):707–710, 2010.

[Mattei *et al.*, 2015] Nicholas Mattei, Judy Goldsmith, Andrew Klapper, and Martin Mundhenk. On the complexity of bribery and manipulation in tournaments with uncertain information. *Journal of Applied Logic*, 13(4):557–581, 2015.

[Mazar *et al.*, 2008] Nina Mazar, On Amir, and Dan Ariely. The dishonesty of honest people: A theory of self-concept maintenance. *Journal of marketing research*, 45(6):633–644, 2008.

[Parkes and Xia, 2012] David C Parkes and Lirong Xia. A complexity-of-strategic-behavior comparison between schulze's rule and ranked pairs. In *AAAI*, 2012.

[Wojtas and Faliszewski, 2012] Krzysztof Wojtas and Piotr Faliszewski. Possible winners in noisy elections. In *AAAI*, 2012.

[Xia, 2012] Lirong Xia. Computing the margin of victory for various voting rules. In *ACM EC*, pages 982–999, 2012.