Election with Bribe-Effect Uncertainty: A Dichotomy Result

Abstract

We consider the electoral bribery problem in computational social choice. Extensive researches have been carried out to analyze the computational vulnerability of different voting rules. Essentially all prior works assume a deterministic model, in which every voter has a specific threshold for bribery. If the briber pays an amount of money above this threshold, the voter will change his/her mind and votes according to the briber. Otherwise the voter refuses to change his/her mind. In this paper, we consider a more realistic setting where each voter has a willness function instead of a fixed threshold. The willness function characterizes how likely the voter will change his/her mind for every amount of bribe. For example, if the briber spends more money, the voter may be more willing to take the bribe. We characterize the computational complexity of the bribery problem where each voter is associated with such a willingness function.

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.* [2019]). One of the most fundamental problems in computational social choice is the *bribery* problem, where an attacker (i.e., briber) attempts to manipulate the outcome of an election by bribing some voters to deviate from their own preference to the attacker's preference. Since its introduction by Faliszewski *et al.* [2009a], this problem has received a considerable amount of attention; see, e.g., Lin [2010]; Brelsford *et al.* [2008]; Xia [2012]; Faliszewski *et al.* [2015, 2011, 2009b]; Parkes and Xia [2012]; Faliszewski *et al.* [2019]; Chen *et al.* [2019].

In this paper, we introduce and investigate a new bribery problem, which differs from the ones that have been studied in the literature as follows. Existing studies essentially assume that a bribed voter makes a *binary* decision in the following sense: declining a bribe that is below a threshold value determined by the voter, and taking a bribe that equals or exceeds the threshold value. This binary decision assumption oversimplifies the problem because in the real world, whether a voter will take a bribe or not would have some inherent uncertainty. For example, if a voter would take a bribe worth \$100, which may be interpreted as the threshold value mentioned above, the voter may also decide to take a bribe worth \$99 with some probability. That is, the *willingness* that a voter will change mind for a bribe may not be a binary decision based on the threshold value and the amount of bribe, as is observed by researchers in the field of psychology (see, e.g., Gerlach *et al.* [2019]). Indeed, Frank and Schulze [2000] noticed that it is far from sufficient to model voters as machines that return "yes" or "no" based on whether or not a monetary award exceeds a threshold. However, most of the prior researches are carried out in the field of psychology. We are among the first researchers who take into account of this critical fact in computational social choice.

The preceding discussion leads to the formulation of the following new problem, Bribery with Bribe-Taking Uncertainty (BDMU). In this problem, each voter v_j is associated with a function f_j, which describes how the amount of bribe to voter v_j affects v_j's decision in taking the bribe or not. More specifically, the *willingness function* $f_j : \mathbb{R}_{\geq 0} \to [0,1]$ maps an amount of bribe to the probability that voter v_j will take the bribe and therefore vote according to the attacker's pereference. For example, $f_j(100) = 0.5$ means that voter v_j, once offered a bribe of 100, may vote according to the attacker's pereference with probability 0.5. Here the probability characterizes the likelihood that a voter will change his/her mind. If, say, the briber tries to bribe a large number of independent voters such that each of them has a probability of 0.5 to change his/her preference, then it is almost certain (with very high probability) that eventually half of the voters will change their preferences. The research question is: Can an attacker with a *fixed bribery budget* succeed in manipulating an election with a *probability exceeding a given threshold*?

We remark that when we say the briber stays within his/her budget, it is actually a bit controversial regarding whether or not the cost has been spent if a voter refuses to change his/her preference. In many situations, there is no guarantee whether a voter may vote according to the briber even if he/she takes the bribe. In case of lobbying, a briber may hire a lobbyist who tries to persuade a voter to change his/her decision, which incurs a cost even if the lobbying fails. Therefore, throughout this paper we assume that a cost is always incurred when the briber decides to bribe a voter, no matter this voter changes his/her prefernce or not.

1.1 Our Contributions

The conceptual contribution of the paper is the introduction of a new type of uncertainty into election models. This uncertainty is described by a willingness function, which goes much beyond the aforementioned binary decision function widely assumed in the literature. The classical problem with a binary decision function, namely $f_j(x) = 0$ for $x \leq \theta_j$ and $f_j(x) = 1$ otherwise (where θ_j is some fixed threshold called the bribing cost of a voter), is a special case of the BDMU model with general $f_j : \mathbb{R}_{\geq 0} \to [0,1]$.

The technical contribution of the paper is in two-fold. On one hand, we show that BDMU under the *plurality* voting rule does not admit any $O(1)$-approximation FPT-algorithm for arbitrary willingness functions, assuming FPT$\neq$W[1], implying that election under the bribery-taking uncertainty is computationally resistant to manipulation. On the other hand, we show that if the willingness functions are all "smooth" (more precisely, the logrithm of each willingness function is Lipschitz continuous), then there exists an FPT algorithm that produces $(1+\varepsilon)$-approxmate solution, implying that the election under the bribery-taking uncertainty becomes vulnerable to manipulation attacks.

1.2 Related work

Uncertainty is inherent to many real-world complex problems and therefore understanding and coping with it has become a fundamental research problem in various disciplines. Putting into the context of election, uncertainty is inherent to the bribery problem because it has been noticed in the literature (see Frank and Schulze [2000]) that it is far from sufficient to model voters as machines that return "yes" or "no" based on whether or not a monetary award exceeds a threshold. However, the type of uncertainty we study in the present paper has not been investigated in the literature. The most closely related prior works are Chen *et al.* [2019] and Wojtas and Faliszewski [2012], where voters still have a fixed threshold to be bribed, but they have a "no-show" probability in the sense that a bribed voter may not vote at all. We stress that, the uncertainty considered in Chen *et al.* [2019] and Wojtas and Faliszewski [2012] are completely different from this paper. Chen *et al.* [2019] and Wojtas and Faliszewski [2012] still follow the classical assumption that a voter will be bribed deterministically if and only if the attacker spends an amount of money that exceeds a fixed threshold. The uncertainty involved in their model is that a bribed voter have a probability of not voting. In our model, there is no deterministic threshold anymore – the willingness function characterizes the relationship between the amount of money spent and succeesful rate of the bribery. This is also confirmed by the fact that the problem studied by Chen *et al.* [2019] always denies an $O(1)$-approximation FPT algorithm, while the approximability of our problem is crucially dependent on the "smoothness" of the willingness functions.

Other kinds of uncertainty in the bribery problem were studied before by, e.g., Erdelyi *et al.* [2014]; Mattei *et al.* [2015]; Erdélyi *et al.* [2009].

We stress that we focus on investigating the impact of the property of the willingness function, rather than its specific form, which is actually an open problem that remains to be tackled. Indeed, some researchers argue that a larger "reward" (i.e., bribe in the context of the present paper) would increase the chance of dishonest behavior (see Conrads *et al.* [2014]; Gneezy [2005]); other researchers 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 other researchers argue that they are relatively independent (see Abeler *et al.* [2016]; Fischbacher and Föllmi-Heusi [2013]; Kajackaite and Gneezy [2017]). However, our focus is not to argue what is a more reasonable willingness function; instead, we focus on what property of the willingness function would have what kinds of consequences to the election problem under the type of uncertainty we introduce. Indeed, we show that it is not the *monotonicity* of the willingness function that matters most in determing whether the election problem under the particular type of uncertainty we introduce is vulnerable to manipulation attack or not; in contrast, it is the *smoothness* of the willingness function that matters most and that actually leads to a dichotomy in the vulnerability of the problem. To the best of our knowledge, this perspective is little understood, if not the first time found to be relevant.

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. Note that we do not consider the case that multiple candidates receive the same highest number of votes, in which case a tie-breaker is needed (e.g., coing-flipping).

Bribery problem. In this problem (see, e.g., Faliszewski *et al.* [2009a]; Lin [2010]; Brelsford *et al.* [2008]; Xia [2012]; Faliszewski *et al.* [2015]; Parkes and Xia [2012]; Chen *et al.* [2019]), 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.

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.*

Proof. By definition of approximation algorithm, we know that c_2 also wins with probability 0 in the optimal solution. Suppose on the contrary that the 2-dimensional knapsack instance admits a feasible solution S', then $|S'| = k$ and $\sum_{j \in S'}(a_j, b_j) \leq (A, B)$. Then, we let the attacker bribe the subset of k key voters corresponding to the k items in S' with an amount $\frac{A+a_j}{(k+1)A}$, respectively. It is straigtforward to see that the overall cost is at most 1, which lies within the budget limit. Furthermore, c_2 wins when each of these k voters votes c_2 (i.e., the attacker's designated candidate), which happens with probability $M^{-\sum_{j \in S'} b_j} > 0$; this contradicts the fact that c_2 wins with 0 probability even in the optimal solution. Hence, the claim holds. □

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

Proof. Note that since the attacker has a total budget 1, we have $\sum_{j:v_j \in V'}(A + a_j - \delta) \leq (k+1)A$. Since $|V'| = k$, we know $\sum_{j:v_j \in V'} a_j \leq A + k\delta$. Because $k\delta < 1$ and A and each a_j are all integers, we know $\sum_{j:v_j \in V'} a_j \leq A$. Thus, the claim holds. □

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

Proof. First, if the attacker spends more budget than $\frac{A+a_j}{(k+1)A}$ on bribing, then we can instead let the attacker spend exact budget $\frac{A+a_j}{(k+1)A}$ on bribing because $f_j(x)$ is constructed to be fixed when $x \in [\frac{A+a_j}{(k+1)A}, 1]$ and the attacker cannot spend more than 1. Second, we claim that the attacker spends at least $\frac{A+a_j-\delta}{(k+1)A}$ for $v_j \in V'$. To see this, we observe that in order for c_2 to win, each of the k voters in V' should vote for the attacker's designated candidate. If the attacker spends less than $\frac{A+a_j-\delta}{(k+1)A}$ for bribing, c_2 wins with probability 0, leading to contradiction. Third, we claim that if the attacker spends an amount x_j on bribing v_j, where $\frac{A+a_j-\delta}{(k+1)A} \leq x_j < \frac{A+a_j}{(k+1)A}$, we can change this bribe amount to $\frac{A+a_j}{(k+1)A}$. To see this, we observe that $\sum_{j:v_j \in V'} \frac{A+a_j-\delta}{(k+1)A} \leq \sum_{j:v_j \in V'} x_j \leq 1$, therefore we have $\sum_{j:v_j \in V'} a_j - k\delta \leq A$. Given that the a_j's and A are integers and that $k\delta < 1$, we have $\sum_{j:v_j \in V'} a_j \leq A$. Hence, changing the amount of bribe does not exceed the budget, and the claim holds. □

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}.*

Proof. Consider the following feasible solution V' to the EBEU instance. Let the attacker bribe the key voters according to the feasible solution of the 2-dimensional knapsack instance, and let the attacker bribe each of these key voters, v_j, with an amount $\frac{A+a_j}{(k+1)A}$. It is straightforward to see that the overall cost is at most 1. Candidate c_2 wins if all of the bribed voters indeed vote for c_2, which happens with probability $\prod_{j:v_j \in V'} M^{-b_j} \geq M^{-B}$ because $\sum_{j:v_j \in V'} b_j \leq B$. The preceding argument implies that there exists a feasible solution to the EBEU instance with an objective value at least M^{-B}, while noting that the optimal solution also has an objective value at least M^{-B}. Thus, the α-approximation algorithm to the EBEU instance returns a feasible solution with an objective value at least $1/\alpha \cdot M^{-B}$. Let V^* be the approximation solution, then

$$\prod_{j:v_j \in V^*} M^{-b_j} \geq 1/\alpha \cdot M^{-B}.$$

Since $M > \alpha$, we have $\sum_{j:v_j \in V^*} b_j < B+1$. Since the b_j's and B are all integers, we know $\sum_{j:v_j \in V^*} b_j \leq B$. Hence, the objective value of the α-approximation solution 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}.*

Proof. If the 2-dimensional knapsack instance does not admit any feasible solution, then we know that for any subset S' of k items, either $\sum_{j \in S'} a_j > A$ holds or $\sum_{j \in S'} b_j > B$ holds. By the one-to-one correspondence between the key voters and the items and Claim 3, we know that for any feasible solution V' to the EBEU instance that satisfies $\sum_{j:v_j \in V'} \frac{A+a_j}{(k+1)A} \leq 1$, we have $\sum_{j:v_j \in V'} b_j \geq B+1$, i.e., $\prod_{j:v_j \in V'} M^{-b_j} \leq M^{-B-1}$. Hence, the objective value of the α-approximation solution to the EBEU instance is also bounded by 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.

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

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

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

Proof. By the definition of Lipschitz continuity, we know that for $x \in F_j^+ \cap [0,1]$, there exists some universal constant α_0 such that

$$|\ln f_j((1 \pm \varepsilon)) - \ln f_j(x)| \leq \alpha_0 \varepsilon x \leq \alpha_0 \varepsilon.$$

Consequently, we have

$$e^{-\alpha_0 \varepsilon} \leq \frac{f_j((1 \pm \varepsilon))}{f_j(x)} \leq e^{\alpha_0 \varepsilon}.$$

Note that for a small ε, we have $1 + 1/2 \cdot \alpha_0 \varepsilon \leq e^{\alpha_0 \varepsilon} \leq 1 + 2\alpha_0 \varepsilon$. Thus, Lemma 1 holds. □

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 $f_j(x') = \sup_{t \leq x} f_j(t)$. Consequently, we define $\bar{f}_j$ as:

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

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*

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

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

Proof. We only prove $|\phi_j((1+\varepsilon)x)-\phi_j(x)| \le O(\varepsilon)\phi_j(x)$ because inequality $|\phi_j((1-\varepsilon)x)-\phi_j(x)| \le O(\varepsilon)\phi_j(x)$ can be proved in the same fashion. First, we observe that $\phi_j((1+\varepsilon)x) \ge \phi_j(x)$ and thus we only need to show $\phi_j((1+\varepsilon)x) \le (1+O(\varepsilon))\phi_j(x)$.

Consider $\phi_j((1+\varepsilon)x) = \sup_{t\le(1+\varepsilon)x} f_j(t)$. There are two possibilities. If the largest value is taken at some $t \le x$, then $\phi_j((1+\varepsilon)x) = \phi(x)$ and Lemma 2 is proved; otherwise, we have $\phi_j((1+\varepsilon)x) \le (1+O(\varepsilon))f_j(t)$ for some $x \le t \le (1+\varepsilon)x$. In the latter case, we know that for any such t, Lemma 1 implies $f_j(t) \le (1+O(\varepsilon))f_j(x) \le (1+O(\varepsilon))\phi_j(x)$, which implies $\phi_j((1+\varepsilon)x) \le (1+O(\varepsilon))^2\phi_j(x)$ and thus Lemma 2 is proved. □

From now on we only need to focus on $\phi_j(x)$ instead of $f_j(x)$ because the monontonicity 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 $x, y \in F_j^+ \cap [0,1], x < y$.

Proof. We only prove $\frac{\phi_j(y)}{\phi_j(x)} \le (\frac{y}{x})^{O(1)}$ because the other inequality can be proved in the same fashion. By Lemma 1, we have $\phi_j((1+\varepsilon)z) \le (1+\beta\varepsilon)f(z)$ for some constant β. Since $y = (1+\varepsilon)^{O(1/\varepsilon \log \frac{y}{x})}x$, we have

$$\begin{aligned}
\phi_j(y) &\le (1+\beta\varepsilon)^{O(1/\varepsilon \log \frac{y}{x})}\phi_j(x) \\
&= [(1+\beta\varepsilon)^{\frac{1}{\beta\varepsilon}}]^{O(\beta \log \frac{y}{x})}\phi_j(x) \\
&\le (\frac{y}{x})^{O(\beta)}\phi_j(x) = (\frac{y}{x})^{O(1)}\phi_j(x).
\end{aligned}$$

The corollary holds. □

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 $h \leq O(k/\varepsilon \cdot \log(k/\varepsilon))$, 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 cannidate 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 $1 \leq j \leq n$, 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

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

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 $\lambda \geq 1$, then we have $\pi_2 \leq \lambda^k \pi_1$.*

Proof. Let $V^1 = \{v_{j_1}, v_{j_2}, \cdots, v_{j_k}\}$ and $V^2 = \{v_{\ell_1}, v_{\ell_2}, \cdots, v_{\ell_k}\}$ such that $\sigma(j_h) = \ell_h$ for $1 \leq h \leq k$. Let $Y_{j_h} \in \{0,1\}$ and $Z_{\ell_h} \in \{0,1\}$ be binary random variables that indicate whether or not v_{j_h} and v_{ℓ_h} will vote for the attacker's designated candidate in the two solutions, respectively. Then, we know $\Pr(Y_{j_h} = 1) = \phi(x_{j_h})$ and $\Pr(Z_{\ell_h} = 1) = \phi(x'_{\ell_h})$, and consequently $\Pr(Z_{\ell_h} = 1) \leq \lambda \Pr(Y_{j_h} = 1)$.

Without loss of generality, we assume $\Pr(Z_{\ell_h} = 1) \geq \Pr(Y_{j_h} = 1)$ because if this inequality does not hold, we can simply increase the probability $\Pr(Z_{\ell_h} = 1)$ to make it true. Note that when $\Pr(Z_{\ell_h} = 1)$ increases, voter v_{ℓ_h} will vote for c_m with a higher probability, and consequently π_2 will increase, say, to π'_2. If $\pi'_2 \leq \lambda^k \pi_1$, then we have $\pi_2 \leq \lambda^k \pi_1$. Therefore, assuming $\Pr(Z_{\ell_h} = 1) \geq \Pr(Y_{j_h} = 1)$ is without loss of generality.

Consider Sol_1. When each Y_{j_h} (Z_{ℓ_h}) takes a fixed value, which is either 0 or 1, we call it a *scenario* of the first (second) solution. In each scenario, c_m may or may not win. Now we consider the set of scenarios in the second solution where c_m wins, and let it be $\Omega^2 = \{\omega_1^2, \omega_2^2, \cdots, \omega_u^2\}$ for some u. Let the value of Z_{ℓ_h} be $z_{\ell_h}(\omega_s^2)$ in scenario ω_s^2, then we know that the probability c_m wins in the second solution is:

$$\Pr(\Omega^2) = \sum_{s=1}^{u} \Pr(\omega_s^2) = \sum_{s=1}^{u} \prod_{h=1}^{k} \Pr(Z_{\ell_h} = z_{\ell_h}(\omega_s^2)).$$

For each scenario ω_s^2 in Sol_2, we consider a specific scenario in Sol_1 where random variable Y_{j_h} takes the same value as Z_{ℓ_h}, and denote this scenario by ω_s^1. Denote by Ω^1 the set of these scenarios, then it is straightforward to see that

$$\Pr(\Omega^1) = \sum_{s=1}^{u} \Pr(\omega_s^1) = \sum_{s=1}^{u} \prod_{h=1}^{k} \Pr(Y_{j_h} = z_{\ell_h}(\omega_s^2)).$$

Note that we have $\Pr(Z_{\ell_h} = 1) \leq \lambda \Pr(Y_{j_h} = 1)$ and that by $\Pr(Z_{\ell_h} = 1) \geq \Pr(Y_{j_h} = 1)$ we also have $\Pr(Z_{\ell_h} = 0) \leq \Pr(Y_{j_h} = 0) \leq \lambda \Pr(Y_{j_h} = 0)$. Thus, we have $\Pr(\Omega^2) \leq \lambda^k \Pr(\Omega^1)$. Note also that voters v_{j_h} and v_{ℓ_h} are in the same group V_i. If we compare the two scenarios ω_s^2 and ω_s^1, we know that there are the same number of voters in each V_i that are bribed to vote for the attacker's designated candidate. Consequently, c_m winning in ω_s^2 implies that c_m also wins in ω_s^1. Therefore, Ω^1 consists of the scenarios where c_m wins, and we have

$$\pi_2 = \Pr(\Omega^2) \leq \lambda^k \Pr(\Omega^1) \leq \lambda^k \pi_1.$$

□

Now we are ready to prove Theorem 4.

Proof of Theorem 4. Recall that we can guess the number of bribed voters in all of the V_i's via k^m enumerations. Consider the $\phi_j(1)$'s. For every V_i, we pick k_i voters from V_i whose $\phi_j(1)$'s are the largest, and let the attacker spend a budget of amount $1/k$ to bribe each of these voters. Now we compare this approximate solution with the optimal solution. We can claim that the optimal solution is $k^{O(1)}$ bounded by the approximate solution. If the claim is true, then according to

Lemma 5, we have $\pi^* \le k^{O(k)}\pi'$ where π' is the objective value of the approximate solution, and Theorem 4 is proved.

What remains to be done is to prove the claim mentioned above. Let V^* and V' be the subset of voters that are selected by the optimal solution and the approximate solution, respectively. Let v_{j_h}, where $h \le |V_i \cap V^*|$, be the voters belonging to $V_i \cap V^*$ such that $\phi_{j_h}(1) \ge \phi_{j_{h+1}}(1)$. Let v_{ℓ_h}, where $h \le |V_i \cap V'|$, be the voters belonging to $V_i \cap V'$ such that $\phi_{\ell_h}(1) \ge \phi_{\ell_{h+1}}(1)$. Since the approximate solution has selected voters in V_i that have the the largest $\phi_j(1)$, we have $\phi_{j_h}(1) \le \phi_{\ell_h}(1)$ for all h. Let $x^*_{j_h}$ be the budget the attacker spends to bribe voter v_{j_h} in the optimal solution. According to Corollary 1, we have

$$\phi_{\ell_h}(1/k) \ge \frac{1}{k^{O(1)}}\phi_{\ell_h}(1) \ge \frac{1}{k^{O(1)}}\phi_{j_h}(1) \ge \frac{1}{k^{O(1)}}\phi_{j_h}(x^*_{j_h}).$$

Hence, the claim is true and Theorem 4 is proved. □

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 $1 \le j \le n$, 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. Recall that with $O(k^{mk}/\varepsilon^{mk} \cdot \log^{mk}(k/\varepsilon))$ enumerations, we can guess the *package* of V_i for every i in the optimal solution, where a *package* specifies k_i numbers that are respectively the budget the attacker spends to bribe the k_i selected voters in the optimal solution. We sort these k_i numbers in an arbitrary order, and use a k_i-dimensional binary vector $\tau^i = (\tau^i_1, \tau^i_2, \cdots, \tau^i_{k_i}) \in \{0,1\}^k$ to denote a *partial package*, where $\tau^i_h = 1$ indicates that the h-th element is included in this partial package, and $\tau^i_h = 0$ otherwise. A partial package is used to record the amount of budget that has been spent within the package during the dynamic programming algorithm that will be presented later.

Using Theorem 4, we can find a rantional number q such that $q \le \pi^* \le k^{O(k)}q$. Let $\delta = \varepsilon/k^{O(k)}$ such that $mkq \le \varepsilon\pi^*$. A number in the form of $h\delta$ for some integer $h \ge 0$ is called a *scaled number*.

We define a mega-scenario. As mentioned before, we use a binary random variable to indicate whether or not a bribed voter votes for the attacker's designated candidate, and each scenario represents a possibility that every random variable takes a fixed value (0 or 1). A mega-scenario, represented by an m-dimensional vector $(\Delta_1, \Delta_2, \cdots, \Delta_m)$, denotes the subset of scenarios where there are exactly Δ_i voters in V_i who vote for the attacker's designated candidate. Let Ω_Δ be the set of all mega-scenarios in which c_m wins. It is straightforward to see that Ω_Δ is independent of the solution. For any solution Sol' with an objective value π', we have

$$\pi' = \sum_{\Delta \in \Omega_\Delta} \prod_{i=1}^m \Pr(E_{Sol'}(\Delta_i)),$$

where $E_{Sol'}(\Delta_i)$ is the event that there are exactly Δ_i voters in $V' \cap V_i$ that vote for the attacker's designated candidate. In order to compute π', it suffices to compute $\Pr(E_{Sol'}(\Delta_i))$ for all $0 \le \Delta_i \le k$. Let $\overline{Sol}^*$ be the *well-structured near optimal solution*. If we can find a feasible solution Sol' such that

$$\Pr(E_{Sol'}(\Delta_i)) \ge \Pr(E_{\overline{Sol}^*}(\Delta_i)) - k\delta, \forall 1 \le i \le m$$

then it follows that

$$\begin{aligned}
\pi' &\ge \sum_{\Delta \in \Omega_\Delta} \prod_{i=1}^m \big(\Pr(E_{\overline{Sol}^*}(\Delta_i)) - k\delta\big) \\
&\ge \sum_{\Delta \in \Omega_\Delta} \left(\prod_{i=1}^m \Pr(E_{\overline{Sol}^*}(\Delta_i)) - k\delta\right) \\
&\ge \sum_{\Delta \in \Omega_\Delta} \prod_{i=1}^m \Pr(E_{\overline{Sol}^*}(\Delta_i)) - |\Omega_\Delta| \cdot k\delta \\
&\ge \pi^* - O(\varepsilon).
\end{aligned}$$

Consequently, Theorem 5 is proved given that Sol' is a feasible solution.

Now, it suffices to design an algorithm Sol' which, based on the guessed *package*, selects a subset $V'_i \subseteq V_i$, $|V'_i| = k_i$ such that for any $0 \le h \le k$, the probability that exactly h voters in V'_i vote for the attacker's designated candidate is at least $\Pr(E_{\overline{Sol}^*}(h)) - k\delta$. In what follows we design such an algorithm based on dynamic programming.

We sort voters in V_i in an arbitrary order. We define a *state*, which is a $(2 + k_i + k)$-dimensional vector that consists of three parts $(\gamma, \xi, \tau^i, \rho)$, where $0 \le \xi \le \gamma \le |V_i|$, τ^i is a *partial package*, and $\rho = (\rho_0, \rho_1, \rho_2, \cdots, \rho_k)$ is a $(k+1)$-dimensional vector with ρ_i being a *scaled number*. We define a *feasible* state recursively as follows:

- Initially, $(0, 0, 0, \cdots, 0)$ is defined to be a feasible state.
- Suppose we have determined all the feasible states $(\gamma, \xi, \tau^i, \rho)$ where $\gamma \le \gamma_0$, we say a state $(\gamma+1, \xi', \tau^{i'}, \rho')$ is *feasible* if and only if there exists some *feasible state* $(\gamma, \xi, \tau^i, \rho)$ such that one of the following is true:

Case 1. $\xi' = \xi$, $\tau^{i'} = \tau^i$ and $\rho' = \rho$.

Case 2. $\xi' = \xi + 1$, $\tau^{i'}$ and τ^i only differ at one coordinate, and the value of this coordinate in vector $\tau^{i'}$ is 1 and the value of this coordinate in vector τ^i is 0. The value of ρ and ρ' satisfy the following: Let the coordinate with different values be at the h-th position. By definition of *partial package*, the h-th coordinate represents a certain amount of budget. Let the attacker spend this amount of budget to bribe the $(\gamma+1)$th voter in V_i, and let p be the probability that this voter votes for the attacker's designated candidate. Define $\hat{\rho}_0 = \rho_0(1-p)$ and $\hat{\rho}_h = \rho_{h-1}p + \rho_h(1-p)$ for $h \ge 1$. Then, ρ'_h is the value of $\hat{\rho}_h$ when rounded down to the nearest scaled number.

Consider any partial solution satisfying the following: (i) it selects a subset of ξ voters among the first γ voters in V_i; (ii) it spends budget in such a way that the amount that is spent to bribe voters in V_i coincides with the partial package τ^i; and

(iii) among the selected voters, the probability that exactly h of them vote for the attacker's designated candidate is $\bar{\rho}_h$. Under this circumstance, we claim that there must exist some *feasible state* $(\gamma, \xi, \tau^i, \rho)$ such that $\rho_h \geq \bar{\rho}_h - \xi\delta$.

We prove the preceding claim by induction on γ. For $\gamma = 0$, the claim is clearly true. Suppose the claim is true for every $\gamma \leq \gamma_0$, now we show that it is also true for $\gamma = \gamma_0 + 1$. Indeed, for any *partial solution* on the first $\gamma_0 + 1$ voters, it either contains the $(\gamma_0 + 1)$the voter or does not contain the $(\gamma_0 + 1)$the voter. If it does not contain the $(\gamma_0 + 1)$th voter, the claim is clearly true. If it does contain the $(\gamma_0 + 1)$th voter, then let $\bar{\rho}_h$ be the probability that there are h voters among the first γ_0 voters that vote for the attacker's designated candidate, and $\bar{\rho}'_h$ be the probability that there are h voters among the first $\gamma_0 + 1$ voters that vote for the attacker's designated candidate. By induction hypothesis, there exists some $(\gamma_0, \xi, \tau^i, \rho)$, where $\rho_h \geq \bar{\rho}_h - \xi\delta$. Meanwhile, we have $\bar{\rho}'_0 = \bar{\rho}_0(1-p)$ and $\bar{\rho}'_h = \bar{\rho}_{h-1}p + \bar{\rho}_h(1-p)$ for $h \geq 1$. By definition, there exists a feasible state $(\gamma_0 + 1, \xi + 1, \tau^{i'}, \rho')$ that is constructed according to case 2. It is easy to verify that $(\gamma_0 + 1, \xi + 1, \tau^{i'}, \rho')$ satisfies $\rho'_h \geq \bar{\rho}_h - (\xi + 1)\delta$. The claim holds.

Since the aforementioned claim is true, by setting $\gamma = |V_i|$ the preceding dynamic programming algorithm can return a subset $V'_i \subseteq V_i$ with $|V'_i| = k_i$ such that for any $0 \leq h \leq k$, the probability that there are exactly h voters in V'_i that vote for the attacker's designated candidate is at least $\Pr(E_{\overline{Sol}^*}(h)) - k\delta$. Thus, Theorem 5 holds. $\square$

5 Conclusion and Discussion

In this paper, we give the first systematic analysis on the bribery problem by taking into account of the uncertainty involved in the decision making of human. We go beyond the classical research that assumes a fixed threshold for voters to accept or decline a bribe. Instead, we use a willingness function that characterizes how human's mind may change according to the amount of bribe. We prove that, interestingly, the computational vulnerability of the election is dependent on whether the logrithm of the willingness function is Lipschitz continuous.

We only consider the Plurality rule in this paper. It would be interesting to consider other rules like veto or Borda count.

References

Johannes Abeler, Daniele Nosenzo, and Collin Raymond. Preferences for truth-telling. 2016.

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

Eric Brelsford, Piotr Faliszewski, Edith Hemaspaandra, Henning Schnoor, and Ilka Schnoor. Approximability of manipulating elections. In *AAAI*, volume 8, pages 44–49, 2008.

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

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

Gábor Erdélyi, Henning Fernau, Judy Goldsmith, Nicholas Mattei, Daniel Raible, and Jörg Rothe. The complexity of probabilistic lobbying. In *International Conference on Algorithmic DecisionTheory*, pages 86–97. Springer, 2009.

Gabor Erdelyi, Edith Hemaspaandra, and Lane A Hemaspaandra. Bribery and voter control under voting-rule uncertainty. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 61–68. International Foundation for Autonomous Agents and Multiagent Systems, 2014.

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

Piotr Faliszewski, Edith Hemaspaandra, Lane A Hemaspaandra, and Jörg Rothe. Llull and copeland voting computationally resist bribery and constructive control. *Journal of Artificial Intelligence Research*, 35:275–341, 2009.

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

Piotr Faliszewski, Yannick Reisch, Jörg Rothe, and Lena Schend. Complexity of manipulation, bribery, and campaign management in bucklin and fallback voting. *Autonomous Agents and Multi-Agent Systems*, 29(6):1091–1124, 2015.

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

Urs Fischbacher and Franziska Föllmi-Heusi. Lies in disguise—an experimental study on cheating. *Journal of the European Economic Association*, 11(3):525–547, 2013.

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

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

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

Agne Kajackaite and Uri Gneezy. Incentives and cheating. *Games and Economic Behavior*, 102:433–444, 2017.

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

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

Andrew Lin. The complexity of manipulating k-approval elections. *arXiv preprint arXiv:1005.4159*, 2010.

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.

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.

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

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

Lirong Xia. Computing the margin of victory for various voting rules. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, pages 982–999. ACM, 2012.