Election with Bribed Voter Uncertainty: Hardness and Approximation Algorithm

Lin Chen,[1] **Lei Xu,**[2] **Shouhuai Xu,**[3] **Zhimin Gao,**[1] **Weidong Shi**[1]
[1]Department of Computer Science, University of Houston, TX, USA
[2]Conduent Labs, NC, USA
[3]Department of Computer Science, University of Texas at San Antonio, TX, USA
chenlin198662@gmail.com, xuleimath@gmail.com, shouhuai.xu@utsa.edu, gao@kell.vin, larryshi@ymail.com

Abstract

Bribery in election (or computational social choice in general) is an important problem that has received a considerable amount of attention. In the classic bribery problem, the briber (or attacker) bribes some voters in attempting to make the briber's designated candidate win an election. In this paper, we introduce a novel variant of the bribery problem, "Election with Bribed Voter Uncertainty" or BVU for short, accommodating the uncertainty that the vote of a bribed voter may or may not be counted. This uncertainty occurs either because a bribed voter may not cast its vote in fear of being caught, or because a bribed voter is indeed caught and therefore its vote is discarded. As a first step towards ultimately understanding and addressing this important problem, we show that it does not admit any *multiplicative* $O(1)$-approximation algorithm modulo standard complexity assumptions. We further show that there is an approximation algorithm that returns a solution with an *additive*-ε error in FPT time for any fixed ε.

Introduction

In multiagent systems, election (or voting) is an important mechanism for collective decision-making. This importance has led to extensive investigations of various aspects of election. Indeed, the field of *computational social choice* investigates algorithmic and computational complexity aspects of this mechanism (see, e.g., the book by Brandt et al. (2016)). In this paper, we focus on two important aspects of election that have received an extensive amount of attention but are still not fully understood: *uncertainty* and *bribery*.

Uncertainty. Most studies in election investigated *deterministic* models and did not consider *uncertainty*, which is however often encountered in real-world scenarios. There are two exceptions. One exception is the investigation of uncertainty from the perspective of the *possible winner*. In this perspective, the input is *incomplete* and the problem is to determine if it is possible to extend the incomplete input to make a designated candidate win or lose. The uncertainty can be incurred by voters' incomplete preference lists, as shown by Konczak and Lang (2005); Xia and Conitzer (2011); Betzler and Dorn (2010); Baumeister and Rothe (2012); Betzler, Hemmann, and Niedermeier (2009). The uncertainty can also be incurred by an incomplete set of candidates (e.g., additional candidates may be added), as shown by Chevaleyre et al. (2010); Xia, Lang, and Monnot (2011); Baumeister, Roos, and Rothe (2011). The other exception is the investigation of uncertainty incurred by complete but *probabilistic* inputs. For example, Wojtas and Faliszewski (2012) introduced an election model in which voters or candidates may have some *probabilities of no-show*, either because the communication network is not reliable or because voters inherently behave as such.

Bribery. Faliszewski, Hemaspaandra, and Hemaspaandra (2009) introduced the *bribery* problem in which a briber (or attacker) attempts to make a designated candidate win by paying a (monetary) bribe to some voters. Once bribed, a voter will vote for the candidate designated by the attacker. This problem has received a considerable amount of attention; see, e.g., Lin (2010); Brelsford et al. (2008); Xia (2012); Faliszewski et al. (2015); Faliszewski, Hemaspaandra, and Hemaspaandra (2011); Faliszewski et al. (2009); Parkes and Xia (2012); Chen et al. (2018a). Most studies in this context consider *deterministic* models, but researchers have started investigating the issue of *uncertainty* in this context as well. For example, Erdelyi, Hemaspaandra, and Hemaspaandra (2014) considered the bribery problem with uncertain voting rules; Mattei et al. (2015) considered the bribery problem with uncertain information, Erdélyi et al. (2009) considered uncertainty in the lobbying problem, which is related to, but different from, the bribery problem.

New problem: Election with Bribed Voter Uncertainty (BVU). We observe that in the context of the bribery problem, there is an inherent uncertainty that has not been considered in the literature: The vote of a bribed voter may or may not be counted, either because a bribed voter may choose not to cast its vote in fear of being caught, or because a bribed voter is indeed caught and therefore its vote is discarded. In this setting, each voter is associated with a price of being bribed as well as a probability that its vote is not counted upon taking a bribe. The goal of the attacker is to bribe a subset of voters such that the total bribing cost does not exceed a given budget, while the probability that a designated candidate wins the election is maximized.

The importance of understanding bribed voter uncertainty cannot be overestimated. This is because, even with the proliferation of anonymous and unregulated cryptocurrencies

(e.g., Bitcoin) that are deemed as ideal for bribery purposes, there is still a possibility that a bribe-taking voter is detected (see Goldfeder et al. (2017)). In the United States, telling a voter whom to vote for is one type of voting fraud and may cause the votes to be discarded (see Heritage Foundation (2018)), as attested by the case that the Wetumpka City Council District 2 election was switched after 8 ballots were ruled (by a judge) to be thrown out (see Arwood (2017)).

Our Contributions

In this paper we make three main conributions. First, we introduce and initiate the study of the BVU problem, which captures a new form of uncertainty in bribery.

Second, we characterize the hardness of the BVU problem and show that the newly captured uncertainty completely changes the complexity of the bribery problem as follows. In the absence of uncertainty, the bribery problem can be solved by a simple greedy algorithm (as shown by Faliszewski, Hemaspaandra, and Hemaspaandra (2009)). In the presence of uncertainty, assuming $P \neq NP$, there is no $O(1)$-approximation algorithm even if there are only two candidates; assuming $W[1] \neq FPT$, there is no $O(1)$-approximation algorithm that runs in FPT time parameterized by r, which is the difference between the number of votes received by the winner and the number of votes received by the designated candidate in the absence of bribery.

Third, despite the strong hardness result mentioned above, we show the existence of an additive ε-approximation FPT algorithm when the number of candidates is a constant. This means that for an arbitrary small $\varepsilon > 0$, there is an algorithm that runs in FPT-time (parameterized by the parameter r mentioned above) and returns an approximate solution with an objective value that is at most ε smaller than the optimal objective value. This result relies on a reduction from the BVU problem to a new variant of the knapsack problem (involving a stochastic objective and multiple cardinality constraints) and an approximation algorithm for this new variant of the knapsack problem (while leveraging dynamic programming and a non-trivial application of Berry-Essen's Theorem). Both the proof technique and the new variant of the knapsack problem may be of independent interest.

All of the omitted proofs can be found in the full version of the present paper Chen et al. (2018b).

Problem Statement and Preliminaries

The basic election model. In the basic election model, there are a set of m candidates $\mathscr{C} = \{c_1, c_2, \ldots, c_m\}$ and a set of n voters $\mathscr{V} = \{v_1, v_2, \ldots, v_n\}$. Each voter has a preference over the candidates. There is a voting *rule* according to which a winner is selected. In this paper we focus on the *plurality rule* with a single winner, namely that every voter votes for its most preferred candidate and the winner is the candidate that receives the highest number of votes.

The classic bribery problem in the basic election model. A voter may be bribed to deviate from its own preference. Suppose each voter v_i has a price q_i. If v_i takes a bribe of amount q_i from the briber (or attacker), then v_i will vote, regardless of v_i's own preference, for the *designated candidate* of the brier (i.e., the candidate preferred by the briber). The briber has a total bribe budget Q. The goal of the briber is to make the designated candidate win the election. The bribery problem has been extensively investigated in the literature; see, for example, Faliszewski, Hemaspaandra, and Hemaspaandra (2009); Lin (2010); Brelsford et al. (2008); Xia (2012); Faliszewski et al. (2015); Parkes and Xia (2012).

BVU (Election with Bribed Voter Uncertainty): A new problem. As discussed before, we introduce and study a novel variant of the classic bribery problem. Suppose voter v_i takes a bribe of amount q_i from the briber. With probability $p_i \in [0,1]$, which is independent of anything else, the vote of v_i goes to the designated candidate and is counted; with probability $1-p_i$, the vote of v_i is *not* counted (for the two reasons mentioned above), that is, *no* candidate will receive the vote from v_i. Without loss of generality, let c_1 be the winner when there is no bribery and c_m be the briber's designated candidate. Let V_j be the subset of voters that vote for candidate c_j in the absence of bribery, then $|V_1| > |V_j|$ for any $j > 1$. Moreover, let $r = |V_1| - |V_m|$, namely the difference between the number of votes received by the winner c_1 and the number of votes received by the designated candidate c_m in the absence of bribery. The BVU problem is formalized as follows, while noting that the voters in V_m do not need to be bribed because they already vote for c_m.

The (Plurality-)BVU Problem
Input: A set of m candidates $\mathscr{C} = \{c_1, c_2, \ldots, c_m\}$, where c_1 is the winner and c_m is the designated candidate in the absence of bribery; a set of n voters $\mathscr{V} = \{v_1, v_2, \cdots, v_n\}$ with $\mathscr{V} = \cup_{j=1}^{m} V_j$, where V_j is the subset of voters that vote for c_j in the absence of bribery; a positive integer $r = |V_1| - |V_m|$; the briber's budget Q; each v_i is associated with a price q_i for bribe and a probability p_i with which the vote of the *bribed* voter v_i goes to the designated candidate c_m and is counted (i.e., $1-p_i$ is the probability that the vote of the *bribed* v_i is not counted)
Output: Find a set of indices $I^* \subseteq \{1, 2, \cdots, n\}$ such that

- $\sum_{i \in I^*} q_i \leq Q$, and
- the probability that the designated candidate c_m wins is maximized by bribing voters in $V' = \{v_i \in V \setminus V_m | i \in I^*\}$

Preliminaries. Let Z be a random variable taking non-negative values. The Markov's inequality (see, for example, Stein and Shakarchi (2009)) says the following: For any $a > 0$, it holds that

$$\Pr(Z \geq a) \leq \frac{\mathbb{E}(Z)}{a}. \tag{1}$$

Theorem 1 (Berry-Essen's Theorem; see Berry (1941))**.** *Let $Z_1, Z_2, \ldots, Z_n$ be independent random variables with $\mathbb{E}(Z_i) = 0$, $\mathbb{E}(Z_i^2) = \sigma_i^2 > 0$, and $\mathbb{E}(|Z_i|^3) = \rho_i < \infty$. Let*

$$S_n = \frac{Z_1 + Z_2 + \ldots + Z_n}{\sqrt{\sigma_1^2 + \sigma_2^2 + \ldots + \sigma_n^2}}.$$

Then, it holds that

$$\sup_{x \in \mathbb{R}} |F_n(x) - \Phi(x)| \leq C_0 \cdot \psi_0, \tag{2}$$

where C_0 is a universal constant, $F_n(x)$ is the cumulative distribution function of S_n, $\Phi(x)$ is the standard normal distribution $\mathcal{N}(0,1)$, and

$$\psi_0 = \left(\sum_{i=1}^{n} \sigma_i^2\right)^{-3/2} \cdot \sum_{i=1}^{n} \rho_i.$$

The following Corollary is a folklore. For completeness, its proof is provided in the full version Chen et al. (2018b).

Corollary 1. *Let Y_j and Z_j for $1 \le j \le m$ be $2m$ independent random variables taking values in $\mathbb{Z}_{\ge 0}$ such that for any integer $0 \le h \le \ell$ and $1 \le j \le m$, the following holds:*

$$\Pr(Y_j \ge h) \ge (1-\delta)\Pr(Z_j \ge h) - \delta.$$

Then, we have:

$$\Pr\left(\sum_{j=1}^{m} Y_j \ge \ell\right) \ge (1-\delta)^m \Pr\left(\sum_{j=1}^{m} Z_j \ge \ell\right) - m\delta.$$

Hardness of the BVU Problem

We show the hardness of the BVU problem for $m = 2$. By introducing *dummy* voters whose prices are higher than the briber's budget Q (i.e., they cannot be bribed), the hardness result immediately applies to the case of an arbitrary $m > 2$.

Hardness Result

The goal of this subsection is to prove the following.

Theorem 2 (Main hardness result)**.** *Assuming $W[1] \neq FPT$, there does not exist an $O(1)$-approximation algorithm for BVU problem that runs in FPT time parameterized by r, even if $m = 2$. Moreover, assuming $P \neq NP$, there does not exist an $O(1)$-approximation algorithm for the BVU problem that runs in polynomial time if r is part of the input, even if $m = 2$.*

In order to prove **Theorem 2**, we leverage the equivalence between the BVU problem with $m = 2$ and the following Knapsack with Uncertainty (KU) problem.

Knapsack with Uncertainty (KU)
Input: A knapsack of capacity Q; a set of n' items, with each item associated with a size q_i and a profit P_i, which is an independent random variable such that $\Pr(P_i = 1) = p_i$ and $\Pr(P_i = 0) = 1 - p_i$; a positive integer r.
Output: Find a set of indices $I^* \subseteq \{1, 2, \cdots, n\}$ such that

- $\sum_{i \in I^*} q_i \le Q$, and
- $\Pr\left(\sum_{i \in I^*} P_i \ge r + 1 - |I^*|\right)$ is maximized.

Lemma 1. *The BVU problem with $m = 2$ is equivalent to the KU problem.*

Proof of Lemma 1. Consider the BVU problem with $m = 2$. Recall that c_1 is the winner in the absence of bribery, c_2 is the designated candidate, $r = |V_1| - |V_2|$, and the problem is to bribe a set $V' = \{v_i \in V_1 | i \in I^*\}$ of voters so that the probability c_2 wins is maximized.

Consider the number of votes received by candidates c_1 and c_2 *after* the briber bribes the voters in V'. For a bribed voter $v_i \in V'$, there are two possibilities:

- The vote of v_i is counted, meaning the number of votes received by candidate c_1 decreases by 1 and the number of votes received by candidate c_2 increases by 1.
- The vote of v_i is not counted, meaning the number of votes received by c_1 decreases by 1 but the number of votes received by c_2 remains the same.

This means that the votes received by candidate c_1 decreases to $|V_1| - |V'|$. Hence, for c_2 to win, it needs at least $|V_1| - |V'| + 1$ votes. Given that c_2 originally receives $|V_2|$ votes, at least $|V_1| - |V'| - |V_2| + 1 = r - |I^*| + 1$ votes from the bribed voters are counted. Let X_i be a binary random variable indicating whether the vote of v_i is counted, then $\Pr(X_i = 1) = p_i$ and $\Pr(X_i = 0) = 1 - p_i$. The probability that at least $r - |I^*| + 1$ votes of the bribed voters are counted is $\Pr\left(\sum_{i \in I^*} X_i \ge r + 1 - |I^*|\right)$. That is, the BVU problem with $m = 2$ essentially asks for an index set I^* such that $\sum_{i \in I^*} q_i \le Q$ and $\Pr\left(\sum_{i \in I^*} X_i \ge r + 1 - |I^*|\right)$ is maximized. This is exactly the KU problem. □

In order to prove **Theorem 2**, we also need:

Theorem 3. *Assuming $W[1] \neq FPT$, there does not exist an $O(1)$-approximation algorithm for the KU problem that runs in FPT time parameterized by r.*

Proof of Theorem 3. We leverage the d-sum problem, which is known to be $W[1]$-hard (see Downey and Fellows (1992)), and show a reduction from the d-sum problem to the KU problem. We first review the d-sum problem.

The d-sum Problem (Downey and Fellows (1995))
Input: s positive integer $x_1, x_2, \cdots, x_s$ and an integer t.
Output: Decide whether or not there exists a subset $E \subseteq \{x_1, x_2, \cdots, x_s\}$ of $|E| = d$ elements such that $\sum_{i: x_i \in E} x_i = t$.

The rest is to show the following reduction: If there is an α-approximation algorithm that solves the KU problem in $f(r)n^{O(1)}$ time for some computable function f and some constant α, then this algorithm can be used to solve the d-sum problem in $f(d)m^{O(1)}$ time. This contradicts the $W[1]$-hardness result of the d-sum problem mentioned above.

The details of the reduction follow. Given an instance of the d-sum problem with s integers $x_1, x_2, \cdots, x_s$, we construct an instance of the KU problem as follows. Let $n' = s$ and $r = 2d - 1$. Construct n' items in the KU problem with $p_i = \Pr(P_i = 1) = 2^{-\omega x_i}$ and $q_i = M - \omega x_i$ for $1 \le i \le n$, where $\omega = \lceil \log_2 \alpha \rceil + 1$ and $M = s\omega \sum_{i=1}^{s} x_i$. Let $Q = dM - \omega t$. We make two claims.

Claim 1. *If the d-sum instance admits a solution, then there exists a solution to the KU problem with an objective value at least $2^{-\omega t}$.*

Proof. Suppose the d-sum problem admits a solution E. Let $I = \{i | x_i \in E\}$ be the index set of items in the solution. We

observe that

$$\sum_{i\in I} q_i = dM - \omega \sum_{i\in I} x_i = dM - \omega t = Q, \text{ and}$$

$$\Pr\left(\sum_{i\in I} P_i \geq d\right) = \Pr\left(P_i = 1, \forall i \in I\right) = \prod_{i\in I} p_i = 2^{-\omega t}.$$

Hence, there exists a solution with an objective value at least $2^{-\omega t}$. Thus, **Claim 1** holds. □

Claim 2. *If the d-sum instance does* not *admit a solution, then any solution to the KU problem has an objective value at most $2^{-\omega(t+1)} < 1/\alpha \cdot 2^{-\omega t}$.*

Proof. Suppose the d-sum problem does not admit a solution. Note that for any solution I to the KU problem, we have $|I| \leq d$; otherwise, $|I| \geq d+1$ leads to

$$\sum_{i\in I} q_i \geq (d+1)M - \omega \sum_{i\in I} x_i > dM > Q,$$

which contradicts that I is a solution. We split $|I| \leq d$ into two scenarios: $|I| \leq d-1$ or $|I| = d$.

- In the case $|I| \leq d-1$, **Claim 2** holds because

$$\Pr\left(\sum_{i\in I} P_i \geq r+1-|I|\right) \leq \Pr\left(\sum_{i\in I} P_i \geq d+1\right)$$
$$= 0 < 2^{-\omega(t+1)}.$$

- In the case $|I| = d$, the fact $\sum_{i\in I} q_i \leq Q$ and $q_i = M - \omega x_i$ and $Q = dM - \omega t$ implies $\sum_{i\in I} x_i \geq t$. Since the d-sum problem does not admit a solution, either $\sum_{i\in I} x_i \geq t+1$ or $\sum_{i:x_i\in I} x_i \leq t-1$. Given that $\sum_{i\in I} x_i \geq t$, we have $\sum_{i\in I} x_i \geq t+1$. Then, **Claim 2** holds because

$$\Pr\left(\sum_{i\in I} P_i \geq d\right) = \prod_{i\in I} p_i = 2^{-\omega\sum_{i\in I} x_i} \leq 2^{-\omega(t+1)}.$$

□

Under **Claims 1**-**2**, we observe that an α-approximation algorithm for the KU problem can be used to solve the d-sum problem as follows:

- In the case the α-approximation algorithm for the KU porblem returns a feasible solution with an objective value that is $\leq 2^{-\omega(t+1)}$, the optimal objective value is at most $\alpha \cdot 2^{-\omega(t+1)} < 2^{-\omega t}$. **Claim 1** implies that the d-sum instance does not admit a feasible solution.
- In the case the α-approximation algorithm for the KU problem returns a feasible solution with an objective value that is $> 2^{-\omega(t+1)}$, **Claim 2** implies that the d-sum instance must admit a feasible solution.

Hence, any α-approximation algorithm for solving the KU problem can be used to solve the d-sum problem. This completes the proof of **Theorem 3**. □

Corollary 2. *Assuming $P \neq NP$, there does not exist an $O(1)$-approximation algorithm for the KU problem that runs in polynomial time if r is part of the input.*

Now we are ready to prove **Theorem 2**.

Proof of Theorem 2. **Lemma 1** shows that the KU problem is equivalent to the BVU problem with two candidates. The hardness of the KU problem is established by **Theorem 3** and **Corollary 2**. Hence **Theorem 2** holds. □

An Approximation Algorithm in FPT time

Having showed that the BVU problem is hard, now we present an approximation algorithm for solving it. The algorithm runs in FPT time for any fixed constant m and any small constant ε. In terms of approximation ratio, our algorithm returns a value that is $\geq \text{OPT} - \varepsilon$, where $\text{OPT} \in [0,1]$ is the optimal objective value. Note that the hardness result given by **Theorem 2** suggests that an additive approximation algorithm is perhaps the best algorithm we can hope for.

Algorithmic Result

Theorem 4 (Main algorithmic result)**.** *For an arbitrary small constant $\varepsilon > 0$, there exists an algorithm for the BVU problem, which runs in $r^{O(mr/\varepsilon)} + n^{O(m^5/\varepsilon^5)}$ time and returns a solution with an objective value no smaller than $\text{OPT} - \varepsilon$, where $\text{OPT} \in [0,1]$ is the optimal objective value.*

In order to prove **Theorem 4**, we need to design an approximation algorithm for the BVU problem. For this purpose, we define a new variant of the Knapsack problem.

The MKU Problem. The MKU problem deals with items that have deterministic sizes but random profits and involves a stochastic objective function, and the goal is to maximize a certain "overflow" probability under the knapsack's volume and cardinality constraints. More specifically, the MKU problem is defined as follows:

Multi-block Knapsack with Uncertainty (MKU)
Input: A knapsack of capacity Q; a set of items $\mathcal{V} = \{v_1, v_2, \cdots, v_n\}$, with each item associated with a size q_i and a profit P_i, which is an independent random variable such that $\Pr(P_i = 1) = p_i$ and $\Pr(P_i = 0) = 1 - p_i$; a partition of the n items into a constant $m \geq 2$ subsets $V_1, V_2, \cdots, V_m$, and a quota Δ_j for each V_j such that $\Delta_j \leq r+1$ for some positive integer r; a positive integer k such that $k \leq r+1$; a positive index $1 \leq j_0 \leq m-1$.
Output: Find a set of indices $I^* \subseteq \{1, 2, \cdots, n\}$ such that

- $\sum_{i\in I^*} q_i \leq Q$,
- $|V(I^*) \cap V_j| \geq \Delta_j$ for all $1 \leq j \leq m-1$ and $j \neq j_0$,
- $|V(I^*) \cap V_{j_0}| = \Delta_{j_0}$,
- $\Pr\left(\sum_{i\in I^*} P_i \geq k\right)$ is maximized,

where $V(I^*) = \{v_i | i \in I^*\}$.

Note that in the preceding definition, we intentionally make the parameters of the MKU problem correspond to the parameters of the BVU problem exactly, because we intend to reduce the number of notations used in this paper (for better readability). That is, parameters n, m, V_j, Q, p_i, q_i, r and I^* in the BVU problem correspond to the same parameters in the MKU problem. We will use the problem context to distinguish the meanings of these parameters. Because of this,

we say an instance of the MKU problem corresponds to an instance of the BVU problem when they have the same set of parameter values.

Now we show that the BVU problem can be solved efficiently by utilizing an algorithm for the MKU problem.

Theorem 5. *Let $\varepsilon > 0$ be an arbitrary small constant. Denote by OPT_{BVU} and OPT_{MKU} the optimal objective value of the BVU problem and the MKU problem, respectively. A feasible solution to the BVU problem with an objective value at least $OPT_{BVU} - \varepsilon$ can be found in $O(rm\Lambda)$ time, where Λ is the time for finding a feasible solution to the corresponding MKU problem with the objective value at least $OPT_{MKU} - \varepsilon$.*

Proof idea. Here we show the proof idea for a weaker version of the theorem, which finds a near optimal solution for BVU in $O(n^{m-1}\Lambda)$ time, and is actually sufficient for proving **Theorem 4**. We observe that after bribing a fixed subset of voters, the total votes received by candidates $1,\ldots,m-1$ are fixed since they always lose some votes from the bribed voters who originally voted for them. Therefore, we can guess the number of bribed voters in each V_j via n^{m-1} enumerations. Suppose we guess the correct value of each Δ_j, where Δ_j voters in V_j are bribed, then the probability that c_m wins is the same as the probability that at least k votes from the bribed voters are counted for some value k such that $k+|V_m| = \max\{|V_j|-\Delta_j, 1 \le j \le m-1\}+1$. The problem now becomes exactly the MKU problem except that $|V(I^*)\cap V_j| \ge \Delta_j$ is replaced with $|V(I^*)\cap V_j| = \Delta_j$. □

Now we show that there is an approximate algorithm for solving the MKU problem.

Theorem 6 (algorithm for solving the MKU problem)**.** *For any arbitrary small constant $\varepsilon > 0$, there exists an algorithm for the MKU problem that runs in $r^{O(mr/\varepsilon)} + n^{O(m^5/\varepsilon^5)}$ time and returns a solution with an objective value that is no smaller than* $\mathrm{OPT} - \varepsilon$*, where* $\mathrm{OPT} \in [0,1]$ *is the optimal objective value in the MKU problem.*

Proof of Theorem 4. By putting **Theorem 5** and **Theorem 6** together, we obtain **Theorem 4**. □

The Proof of Theorem 6

The main difficulty originates from the maximization of a probability involving the sum of random variables, which does not have a simple explicit expression. A natural idea is to approximate the summation of random variables with a Gaussian variable via Berry-Essen's Theorem. However, such an approximation is not always achievable because the condition in Berry-Essen's Theorem does not necessarily hold. Furthermore, even if Berry-Essen's Theorem is applicable, bounding the tail probability of a Gaussian variable together with a set of other constraints required in MKP is still challenging. Figure 1 highlights the proof strategy for overcoming these difficulties.

Specifically, We partition the set of items into *big* and *small* ones based on their probability. Then, we differentiate the case that the optimal solution contains many big items (**Case 1**), which is easily coped with by using Markov's inequality (**Lemma 2**), from the case that the optimal solution does not contain many big items (**Case 2**), whose treatment is much more complicated and proceeds as follows.

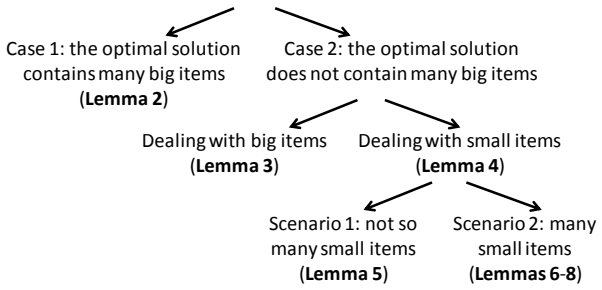

Figure 1: Strategy for proving **Theorem 6**.

- First, we apply **Corollary 1** to decompose the MKU problem in **Case 2** into a series of sub-problems, each of which is a stochastic knapsack problem with one cardinality constraint.
- Second, for big items (**Lemma 3**), we round their probability to $O(k/\varepsilon) \le O(r/\varepsilon)$ distinct probabilities. This allows us to guess the number of big items corresponding to the rounded probabilities in the optimal solution, leading to the selection of the optimal subset of big items.
- Third, for small items (**Lemma 4**), there are two scenarios:
  - In the scenario where the optimal solution does not contain a large volume of small items, we present a dynamic programming algorithm (**Lemma 5**).
  - In the scenario where the optimal solution contains a large volume of small items, Berry-Essen's Theorem is applicable and we can use it to transform the problem of maximizing a specific probability to the problem of approximating the summation of moments of random variables in the optimal solution. Since the moments of a random variable are deterministic, we can leverage the technique for solving the classic knapsack problem (**Lemmas 6-8**).

Definition 1 (big vs. small items)**.** *Under the assumption that $\varepsilon > 0$ is a small constant such that $1/\varepsilon \ge 4$ is an integer, we say an item in the MKU problem is* big *if $p_i > 1-\varepsilon^2$ and is* small *otherwise.*

Lemma 2 (the case the optimal solution containing many big items)**.** *If $|T \cap B_{j^*}| \ge 2k$, then there is a polynomial-time algorithm that returns a solution I to the MKU problem such that*

- $\sum_{i\in I} q_i \le Q$,
- $|V(I)\cap V_j| \ge \Delta_j$ *for* $j \ne j_0$,
- $|V(I)\cap V_{j_0}| = \Delta_{j_0}$, *and*
- $\Pr(\sum_{i\in I} P_i \ge k) \ge 1-\varepsilon$.

Proof idea. We first select the $2k$ big items with the smallest sizes within B_{j^*}. Using Markov's inequality, it is not difficult

to show that even for these $2k$ items, the probability that their total profit is at least k is greater than or equal to $1-\varepsilon$. To further ensure $|V(I)\cap V_j|\geq \Delta_j$, we simply add items of the smallest size in each V_j. □

Lemma 3 (dealing with big items in the case the optimal solution not containing many big items)**.** *If $|T\cap B_\ell|\leq 2k-1$, then there is an algorithm that runs in $k^{O(mk/\varepsilon)}\leq r^{O(mr/\varepsilon)}$ time and returns a set $I\cap B_\ell$ of big items such that*

- $|I\cap B_\ell|=|T\cap B_\ell|$,
- $Q_{I\cap B_\ell}\leq Q_{T\cap B_\ell}$, *and*
- $\Pr(P_{I\cap B_\ell}\geq h)\geq(1-2\varepsilon/m)\Pr(P_{T\cap B_\ell}\geq h)$ *for any* $h\geq 0$.

Proof idea. Note that

$$\Pr(P_{T\cap B}=h)=\sum_{I\subseteq T\cap B,|I|=h}\prod_{i\in I}p_i\prod_{i\notin I}(1-p_i),$$

Since $|T\cap B|<2k$, even if we increase or decrease the value of each p_i in the above equation by a factor of $1+\varepsilon/(mk)$, the value of $\Pr(P_{T\cap B}=h)$ changes by a factor of $1+O(\varepsilon/m)$. Therefore, we can round down the probabilities of all the big items into the form of $(1-\varepsilon^2)(1+\varepsilon/(mk))^j$. By doing so there are only $O(km/\varepsilon)$ different kinds of probabilities. We can guess the number of big items in T that have the same rounded probability, and select the ones with the smallest size instead. □

Lemma 4 (dealing with small items in the case the optimal solution not containing many big items)**.** *There exists an algorithm that runs in $n^{O(m^5/\varepsilon^5)}$ time and returns a feasible solution $I\cap S_\ell$ such that*

- $|I\cap S_\ell|=|T\cap S_\ell|$
- $Q_{I\cap S_\ell}\leq Q_{T\cap S_\ell}$
- $\Pr(P_{I\cap S_\ell}\geq h)\geq\Pr(P_{T\cap S_\ell}\geq h)-\Theta(\varepsilon/m)$

The proof of this lemma needs a sequence of results.

Lemma 5. *For any ζ, there exists an algorithm that runs in $(mn/\varepsilon)^{O(\zeta)}$ time and returns a solution $I\cap S_\ell$ such that $|I\cap S_\ell|=|T\cap S_\ell|$, $Q_{I\cap S_\ell}\leq Q_{T\cap S_\ell}$ and $\Pr(P_{I\cap S_\ell}=h)\leq\Pr(P_{T\cap S_\ell}=h)+2\varepsilon/(mn)$ for every $0\leq h\leq\zeta-1$.*

Proof idea. The algorithm is based on dynamic programming that recursively solves the following sub-problem: Let $S[j]$ be the set of the first j items in S. Find a *partial solution* $I\cap S[j]$ such that $|I\cap S[j]|=|T\cap S[j]|$, $Q_{I\cap S[j]}\leq Q_{T\cap S[j]}$ and $\Pr(P_{I\cap S[j]}=h)\leq\Pr(P_{T\cap S[j]}=h)+j\cdot\varepsilon/(mn^2)$ hold for every $0\leq h\leq\zeta-1$ and $0\leq\varpi\leq n'$. This can be done using a similar approach as that for the classical knapsack problem (see, e.g., Vazirani (2013)), except that we do *not* know the values of $|T\cap S[j]|$, and the value of $\Pr(P_{T\cap S[j]}=h)$ for every j and h. To handle this, we solve sub-problems for all the possible (approximate) values of $|T\cap S[j]|$ and $\Pr(P_{T\cap S[j]}=h)$, that is, for any $0\leq\varpi\leq n'$ and any $u_h\in\{0,\eta,2\eta,\ldots,\eta\cdot 1/\eta\}$ where $\eta=\varepsilon/(mn^2)$, we find a partial solution $I\cap S[j]$ such that $I\cap S[j]=\varpi$, $\Pr(P_{I\cap S[j]}=h)\leq u_h+j\cdot\varepsilon/(mn^2)$ hold for every $0\leq h\leq\zeta-1$. We can further guess (within an error of $O(\eta)$) the value of $\Pr(P_{T\cap S}=h)$ for every h, through which the dynamic programming can return a solution satisfying **Lemma 5**. □

Definition 2. *For any subset D of small items and integer $h\geq 0$, we define*

$$\hat{h}_D=\frac{h-\sum_{i\in D}p_i}{\sqrt{\sum_{i\in D}\sigma_i^2}}=\frac{h-\sum_{i\in D}p_i}{\sqrt{\sum_{i\in D}p_i(1-p_i)}}.$$

The proofs of the following two lemmas mainly consist of mathematical calculations together with a suitable application of Berry-Essen's theorem.

Lemma 6. *If*

$$\sum_{i\in I\cap S_\ell}p_i>(m/\varepsilon)^4\ \text{ and }\ |\Phi(\hat{h}_{I\cap S_\ell})-\Phi(\hat{h}_{T\cap S_\ell})|\leq O(\varepsilon/m),$$

then

$$\Pr\left(\sum_{i\in I\cap S_\ell}P_i\geq h\right)\geq\Pr\left(\sum_{i\in T\cap S_\ell}P_i\geq h\right)-\Omega(\varepsilon/m),$$

where $\Phi(x)$ is the cumulative distribution function of the standard normal distribution.

Lemma 7. *If*

$$\sum_{i\in T\cap S_\ell}p_i>(m/\varepsilon)^4$$

and the following holds for some $I\cap S_\ell$:

- $|\sum_{i\in I\cap S_\ell}p_i-\sum_{i\in T\cap S_\ell}p_i|\leq O(\varepsilon/m)$, *and*
- $|\sum_{i\in I\cap S_\ell}p_i(1-p_i)-\sum_{i\in T\cap S_\ell}p_i(1-p_i)|\leq O(\varepsilon/m)$,

then

$$|\Phi(\hat{h}_{I\cap S_\ell})-\Phi(\hat{h}_{T\cap S_\ell})|\leq O(\varepsilon/m)$$

for every $0\leq h\leq k$.

Now we can replace the condition of $\Pr(P_{I\cap S_\ell}\geq h)\geq\Pr(P_{T\cap S_\ell}\geq h)-\Theta(\varepsilon/m)$ in **Lemma 4** with the conditions in **Lemma 7**, leading to the following lemma whose proof is based on a dynamic programming approach similar to that of **Lemma 5**.

Lemma 8. *If $\sum_{i\in T\cap S_\ell}p_i>(m/\varepsilon)^4$, then there exists an algorithm that runs in $O(m^2n^5/\varepsilon^2)$ time and returns a feasible solution with item set $I\cap S_\ell$ such that*

- $Q_{I\cap S_\ell}\leq Q_{T\cap S_\ell}$, *and*
- $|I\cap S_\ell|=|T\cap S_\ell|$, *and*
- $|\sum_{i\in I\cap S_\ell}p_i-\sum_{i\in T\cap S_\ell}p_i|\leq O(\varepsilon/m)$, *and*
- $|\sum_{i\in I\cap S_\ell}p_i(1-p_i)-\sum_{i\in T\cap S_\ell}p_i(1-p_i)|\leq O(\varepsilon/m)$.

Proof of Lemma 4. Without loss of generality we assume $S_\ell=\{1,2,\cdots,n'\}$. Recall that S_ℓ consists of small items. For any small item v_i we have $\varepsilon^2\leq 1-p_i\leq 1$. We prove **Lemma 4** by considering the following two scenarios:

- **Scenario 1:** $\sum_{i\in T\cap S_\ell}p_i\leq(m/\varepsilon)^4$.
- **Scenario 2:** $\sum_{i\in T\cap S_\ell}p_i>(m/\varepsilon)^4$.

In **Scenario 1**, we observe that by Markov's inequality Eq.(1), we know

$$\Pr(P_{T\cap S_\ell}\geq h)\leq\varepsilon/m\ \text{ for }\ h\geq(m/\varepsilon)^5.$$

Let $\zeta = (m/\varepsilon)^5$. **Lemma 5** showed that we can find a subset $I \cap S_\ell$ of items in polynomial time such that

$$\Pr(P_{I\cap S_\ell} = h) \leq \Pr(P_{T\cap S_\ell} = h) + 2\varepsilon/(mn)$$

holds for every $0 \leq h \leq \zeta - 1$. Then

$$\Pr(P_{I\cap S_\ell} \geq h) \geq \Pr(P_{T\cap S_\ell} \geq h) - 2\varepsilon/m$$

for every $0 \leq h \leq \zeta - 1$. Since

$$\Pr(P_{I\cap S_\ell} \geq h) \geq 0 \geq \Pr(P_{T\cap S_\ell} \geq h) - 2\varepsilon/m$$

for $h \geq \zeta$, we find a near-optimal solution $I \cap S_\ell$ in polynomial time. Hence, **Lemma 4** holds in **Scenario 1**.

In **Scenario 2**, we have $\sum_{i\in T\cap S_\ell} p_i > (m/\varepsilon)^4$. As highlighted before, the difficulty encountered here is to maximize the probability with respect to the sum of random variables. Our strategy is to first replace the condition

$$\Pr(P_{I\cap S_\ell} \geq h) \geq \Pr(P_{T\cap S_\ell} \geq h) - \Theta(\varepsilon/m) \tag{3}$$

in **Lemma 4** with a stronger, but handier condition. More precisely, by **Lemma 6** and **Lemma 7**, we show that Eq (3) is true if we have $|\sum_{i\in I\cap S_\ell} p_i - \sum_{i\in T\cap S_\ell} p_i| \leq O(\varepsilon/m)$, and $|\sum_{i\in I\cap S_\ell} p_i(1-p_i) - \sum_{i\in T\cap S_\ell} p_i(1-p_i)| \leq O(\varepsilon/m)$, plus some cardinality constraints. Note that these knapsack-like constraints are much easier to handle when compared with Eq (3). We will design a dynamic programming based algorithm that finds a feasible solution with respect to these stronger but handier conditions (**Lemma 8**). Thus, **Lemma 4** holds in **Scenario 2**. □

Now we are ready to prove **Theorem 6**.

Proof of Theorem 6. Let T be the set of *indices* of items that are selected by the optimal solution to the MKU problem, and $\text{OPT}_{MKU} = \Pr(\sum_{i\in T} P_i \geq k)$ be the optimal objective value given by the optimal solution. For any I (i.e., the indices of the items that are selected by an approximation algorithm), we define $P_I = \sum_{i\in I} P_i$ and $Q_I = \sum_{i\in I} q_i$. Denote by S the set of *indices* of small items and B the set of *indices* of big items. Let $B_j = \{i | i \in B, v_i \in V_j\}$ and $S_j = \{i | i \in S, v_i \in V_j\}$. According to the number of big items selected by the optimal solution in each V_j, namely $|T \cap B_j|$, we divide the MKU problem into the following two cases:

- **Case 1**: There exists some $1 \leq j^* \leq m$ such that $|T \cap B_{j^*}| \geq 2k$.
- **Case 2**: $|T \cap B_j| \leq 2k-1$ for every $1 \leq j \leq m$.

In **Case 1**, as $1 \leq j^* \leq m$, we can guess j^* by $O(m)$ enumerations. When the guess of j^* is correct, **Theorem 6** is proven as **Lemma 2**.

In **Case 2**, we have $|T \cap B_j| \leq 2k-1$ for every j. We first guess the values of $|T \cap B_j|$ and $|T \cap S_j|$ for all j, leading to $n^{O(m)}$ enumerations. For the correct guess, **Corollary 1** says that a near optimal solution I can be found when the following conditions are satisfied simultaneously:

- $|I \cap B_j| = |T \cap B_j|$ and $|I \cap S_j| = T \cap S_j|$.
- $Q_{I\cap B_j} \leq Q_{T\cap B_j}$ and $Q_{I\cap S_j} \leq Q_{T\cap S_j}$.
- For $\delta = \Theta(\varepsilon/m)$ and any $0 \leq h \leq k$, we have

$$\Pr(P_{I\cap S_j} \geq h) \geq (1-\delta)\Pr(P_{T\cap S_j} \geq h) - \delta, \tag{4a}$$
$$\Pr(P_{I\cap B_j} \geq h) \geq (1-\delta)\Pr(P_{T\cap B_j} \geq h) - \delta. \tag{4b}$$

This means that we can decompose the MKU problem in **Case 2** into a sequence of sub-problems, each of which asks for a near optimal solution $I \cap B_j$ or $I \cap S_j$. Let $1 \leq \ell \leq m$ be an arbitrary index. Then, **Theorem 6** in **Case 2** is proven as **Lemma 3** (dealing with big items) and **Lemma 4** (dealing with small items). □

Conclusion and Discussion

We have introduced the BVU problem with the uncertainty that the vote of a bribed voter may not be counted (either because the bribed voter does not cast its vote in fear of being caught, or because the bribed voter is indeed caught and its vote is discarded). We have showed that the BVU problem does not admit any *multiplicative* $O(1)$-approximation algorithm in FPT time modulo standard complexity assumptions. We have also showed that there is an algorithm that returns an approximate solution with an *additive*-ε error in FPT time for any arbitrary small ε. Given the hardness result, this algorithm is perhaps the best one can hope for.

The BVU problem has many interesting aspects that deserve further studies. First, our algorithmic result assumes a constant number of candidates. Future work needs to characterize the hardness of, and design approximate algorithms for, the BVU problem when the number of candidates is part of the input (rather than a constant). The problem with an arbitrary number of candidates may be strictly harder than that of a constant number of candidates. It is not clear whether or not our approximation algorithm, which works for a constant number of candidates, can be extended to cope with the case of an arbitrary number of candidates. Moreover, the hardness of the BVU problem (with both a constant and an arbitrary number of candidates) needs to be investigated with respect to other voting rules, such as the *k-approval* or *Borda* rule. Nevertheless, our hardness result with $m = 2$ candidates immediately implies a hardness result with respect to the *Borda* voting rule.

Furthermore, the notion of *uncertainty* is a rich topic that needs to be explored further. Even for the particular kind of uncertainty introduced in the present paper, there are many problems that deserve to be studied. For example, it is interesting to incorporate the *probabilistic no-show* introduced by Wojtas and Faliszewski (2012) into our model such that *unbribed* voters have some probabilities of no-show (i.e., not casting their votes); of course, the reason that an unbribed voter may not cast its vote is different from the aforediscussed reason that the vote of a bribed voter may not be counted. Another outstanding future work is to consider the probability that a voter accepts a bribe. Moreover, the literature focuses on the setting where the costs of bribery are deterministic. However, such a cost usually is only an estimation because it is private to a voter. Therefore, it is perhaps more reasonable to assume that the probability that a voter takes a bribe depends on the price offered by the briber.

Acknowledgement. We thank the anonymous reviewers for their constructive comments. Research was supported in part by NSF 1756014.

References

Arwood (2017), C. http://www.thewetumpkaherald.com/2017/08/23/washington-wins-district-2-council-dispute/.

Baumeister, D., and Rothe, J. 2012. Taking the final step to a full dichotomy of the possible winner problem in pure scoring rules. *Information Processing Letters* 112(5):186–190.

Baumeister, D.; Roos, M.; and Rothe, J. 2011. Computational complexity of two variants of the possible winner problem. In *AAMAS*, 853–860. IFAAMAS.

Berry, A. C. 1941. The accuracy of the gaussian approximation to the sum of independent variates. *Transactions of the american mathematical society* 49(1):122–136.

Betzler, N., and Dorn, B. 2010. Towards a dichotomy for the possible winner problem in elections based on scoring rules. *Journal of Computer and System Sciences* 76(8):812–836.

Betzler, N.; Hemmann, S.; and Niedermeier, R. 2009. A multivariate complexity analysis of determining possible winners given incomplete votes. In *IJCAI*, volume 9, 53–58.

Brandt, F.; Conitzer, V.; Endriss, U.; Procaccia, A. D.; and Lang, J. 2016. *Handbook of computational social choice*. Cambridge University Press.

Brelsford, E.; Faliszewski, P.; Hemaspaandra, E.; Schnoor, H.; and Schnoor, I. 2008. Approximability of manipulating elections. In *AAAI*, volume 8, 44–49.

Chen, L.; Xu, L.; Xu, S.; Gao, Z.; Shah, N.; Lu, Y.; and Shi, W. 2018a. Protecting election from bribery: New approach and computational complexity characterization. In *AAMAS 2018*, 1894–1896.

Chen, L.; Xu, L.; Xu, S.; Gao, Z.; and Shi, W. 2018b. Election with bribed voter uncertainty: Hardness and approximation algorithm. *arXiv preprint arXiv:1811.03158*.

Chevaleyre, Y.; Lang, J.; Maudet, N.; and Monnot, J. 2010. Possible winners when new candidates are added: The case of scoring rules. In *AAAI*.

Downey, R. G., and Fellows, M. R. 1992. Fixed-parameter intractability. In *Structure in Complexity Theory Conference, 1992., Proceedings of the Seventh Annual*, 36–49. IEEE.

Downey, R. G., and Fellows, M. R. 1995. Fixed-parameter tractability and completeness ii: On completeness for w [1]. *Theoretical Computer Science* 141(1-2):109–131.

Erdélyi, G.; Fernau, H.; Goldsmith, J.; Mattei, N.; Raible, D.; and Rothe, J. 2009. The complexity of probabilistic lobbying. In *International Conference on Algorithmic DecisionTheory*, 86–97. Springer.

Erdelyi, G.; Hemaspaandra, E.; and Hemaspaandra, L. A. 2014. Bribery and voter control under voting-rule uncertainty. In *AAMAS*, 61–68. International Foundation for Autonomous Agents and Multiagent Systems.

Faliszewski, P.; Hemaspaandra, E.; Hemaspaandra, L. A.; and Rothe, J. 2009. Llull and copeland voting computationally resist bribery and constructive control. *Journal of Artificial Intelligence Research* 35:275–341.

Faliszewski, P.; Reisch, Y.; Rothe, J.; and Schend, L. 2015. Complexity of manipulation, bribery, and campaign management in bucklin and fallback voting. *Autonomous Agents and Multi-Agent Systems* 29(6):1091–1124.

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

Faliszewski, P.; Hemaspaandra, E.; and Hemaspaandra, L. A. 2011. Multimode control attacks on elections. *Journal of Artificial Intelligence Research* 40:305–351.

Goldfeder, S.; Kalodner, H. A.; Reisman, D.; and Narayanan, A. 2017. When the cookie meets the blockchain: Privacy risks of web payments via cryptocurrencies. *CoRR* abs/1708.04748.

Heritage Foundation (2018). https://www.whitehouse.gov/sites/whitehouse.gov/files/docs/pacei-voterfraudcases.pdf.

Konczak, K., and Lang, J. 2005. Voting procedures with incomplete preferences. In *Proc. IJCAI-05 Multidisciplinary Workshop on Advances in Preference Handling*, volume 20.

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

Mattei, N.; Goldsmith, J.; Klapper, A.; and Mundhenk, M. 2015. On the complexity of bribery and manipulation in tournaments with uncertain information. *Journal of Applied Logic* 13(4):557–581.

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

Stein, E. M., and Shakarchi, R. 2009. *Real analysis: measure theory, integration, and Hilbert spaces*. Princeton University Press.

Vazirani, V. V. 2013. *Approximation algorithms*. Springer Science & Business Media.

Wojtas, K., and Faliszewski, P. 2012. Possible winners in noisy elections. In *AAAI*.

Xia, L., and Conitzer, V. 2011. Determining possible and necessary winners given partial orders. *J. Artif. Intell. Res.(JAIR)* 41:25–67.

Xia, L.; Lang, J.; and Monnot, J. 2011. Possible winners when new alternatives join: New results coming up! In *AAMAS*, 829–836. IFAAMAS.

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