Priced Electoral Control Under Constructive Attacks with Uncertainty: Complexity and Algorithm

Abstract

We consider the *electoral control* problem in which an attacker attempts to make a designated candidate of its choice win an election by adding some additional, unregistered voters. More specifically, we initiate the study of the perspective of *uncertainty* in this electoral control problem, where uncertainty means that each unregistered voter has some probability of no-show (e.g., not casting the vote). Since adding an unregistered voter incurs a price or cost to the attacker and the attacker has a limited budget, the problem is to seek optimization as follows. Suppose the designated candidate needs k additional votes from unregistered votes in order to win an election. Because each unregistered voter has some probability of no-show, the attacker's goal is to add a number of unregistered voters to maximize the probability that the designated candidate gets k additional votes (i.e., winning the election), while not violating the constraint imposed by the budget. For this problem, we show that there is no $O(1)$-approximation algorithm that runs in FPT time parameterized by k, unless $W[1] = FPT$, and that there is an additive ε-approximation FPT algorithm that runs in FPT time parameterized by k.

1 Introduction

In multi-agent systems, election or voting is an important mechanism for agents to collectively make decisions. This importance has led to extensive investigations of various aspects of election. Indeed, the field of *computational social choice* investigates the algorithmic and computational complexity aspects of this mechanism (see, e.g., surveys by Faliszewski *et al.* [2010, 2009a] and the references therein).

However, most prior studies investigated *deterministic* models of election without considering *uncertainty*, which is often encountered in real-world scenarios. Exceptions are those studies investigating uncertainty from the perspective of *possible winner*, in which the input is incomplete and the problem is to determine if it is possible to extend the given input to make a designated candidate win or lose. For example, the uncertainty that voters' preference lists are incomplete has been investigated by Konczak and Lang [2005]; Xia and Conitzer [2011]; Betzler and Dorn [2010]; Baumeister and Rothe [2012]; Betzler *et al.* [2009]; the uncertainty that the set of candidates is incomplete (e.g., additional candidates may be added) has been investigated by Chevaleyre *et al.* [2010]; Xia *et al.* [2011]; Baumeister *et al.* [2011].

Another exception, initiated by Wojtas and Faliszewski [2012], is the investigation of the uncertainty that the input is complete but probabilistic. Specifically, they introduced a new election model in which voters or candidates may have some probability of no-show. This kind of uncertainty is interesting because in many multi-agent systems, an agent or voter may only cast its vote with a certain probability either because the communication network is not reliable or because the voter inherently behaves as such.

Wojtas and Faliszewski [2012] further investigated the computational complexity of calculating the candidates' winning probabilities. The difficulty of this computational problem has interesting implications, such as: How difficult is it for an attacker to manipulate elections? Can this difficulty prevent attackers from attempting to manipulate elections? These implications are equally applicable to the general setting of *electoral control*, where an attacker attempts to manipulate elections. The electoral control problem was introduced by Bartholdi *et al.* [1992], which has led to numerous studies.

In this paper, we initiate the study of electoral control with the uncertainty that voters have some probability of no-show. That is, we introduce the perspective of uncertainty pioneered by Wojtas and Faliszewski [2012] into the electoral control problem pioneered by Bartholdi *et al.* [1992]. Because the electoral control problem has many variants, as a first step we focus on studying the incorporation of uncertainty into the specific problem of electoral control with a constructive attacker, which is known as Constructive Control by Adding Voters, or CCAV for short (see, e.g., Brandt *et al.* [2016]). In the CCAV problem, there are a set of candidates, a set of *registered* voters, a set of *unregistered* voters, and a designated candidate of the attacker's choice. The attacker attempts to make the designated candidate win the election by *adding* some unregistered voters in favor of the designated candidate.

More specifically, the uncertainty we introduce into the CCAV problem is that each unregistered voter has some probability of no-show (i.e., an unregistered voter may or may not cast its vote, with some probability). This leads to the problem we call "CCAV with Uncertainty" or CCAV-U for

short. In the CCAV-U problem, each unregistered voter is associated with two parameters: one parameter is a deterministic price or cost incurred to the attacker when adding the unregistered vote, and the other parameter is the probability the unregistered voter will cast its vote once added by the attacker. Suppose the designated candidate needs k additional votes in order to win an election. Suppose the attacker has a limited budget for adding unregistered voters. The question is: Which subset of the unregistered voters should be added in order to maximize the probability that the designated candidate gets k additional votes (i.e., winning the election), while not violating the budget constraint?

1.1 Our contribution

We initiate the study of electoral control with the uncertainty that each unregistered voter has some probability of no-show (i.e., not casting their votes). Since the CCAV-U problem can be specified under various kinds of voting rules, in this paper we focus on the *plurality* voting rule, which will be defined in Section 2. We investigate the hardness and algorithmic aspects of CCAV-U under the plurality voting rule.

We present two results. First, we show that the incorporation of uncertainty completely changes the complexity of the problem. In the absence of uncertainty, the problem is trivial because the attacker can simply add the k unregistered voters that will vote for the designated candidate while incurring the smallest cost. In the presence of uncertainty, however, we show that there is no $O(1)$-approximation algorithm for CCAV-U problem that runs in FPT time parameterized by k, assuming $W[1] \neq FPT$. Second, despite this strong hardness, we show the existence of an additive ε-approximation FPT algorithm, which means that for any constant $\varepsilon > 0$, there is an algorithm that runs in FPT time (parameterized by k) and returns an approximate solution with an objective value that is at most ε smaller than the optimal objective value.

The hardness and algorithmic results mentioned above have the following implications. First, the sharp difference between the complexity of the CCAV-U problem and the complexity of the (deterministic) CCAV problem suggests that new techniques might be needed for coping with the problem of electoral control with uncertainty. We will further discuss this matter in Section 6. Second, although the hardness result of CCAV-U under the plurality voting rule excludes the existence of any $O(1)$-approximation FPT algorithms and seemingly suggests that the corresponding election mechanism is robust against constructive attackers, the existence of an additive ε-approximation algorithm indicates the opposite in a sense. This calls for more studies to deepen our understanding of electoral control with uncertainty (e.g., under what circumstances a hardness result can indeed prevent attackers from manipulating elections).

1.2 Related work

The problem of *electoral control*, which was introduced by Bartholdi *et al.* [1992], has received a large amount of attention and has led to various models (see, e.g., Faliszewski *et al.* [2009b]; Hemaspaandra *et al.* [2007]; Lin [2012]; Erdélyi *et al.* [2011]; Faliszewski *et al.* [2013]; Fitzsimmons *et al.* [2013]; Faliszewski *et al.* [2010] and surveys by Faliszewski *et al.* [2009a]; Brandt *et al.* [2016]). However, these studies focus on different voting rules (e.g., plurality, veto, Borda, etc.) or different ways of control (e.g., adding voters vs. deleting voters). The present study moves a step forward by introducing the perspective of *uncertainty* into the problem of electoral control, while focusing on the specific problem of constructive electoral control under the plurality voting rule.

To the best of our knowledge, uncertainty has been investigated primarily in the context of the *possible winner* problem (see, e.g., Konczak and Lang [2005]; Xia and Conitzer [2011]; Betzler and Dorn [2010]; Baumeister and Rothe [2012]; Betzler *et al.* [2009]; Chevaleyre *et al.* [2010]; Xia *et al.* [2011]; Baumeister *et al.* [2011, 2012]; Hazon *et al.* [2012]; Chevaleyre *et al.* [2010]; Boutilier *et al.* [2014]). Another kind of uncertainty, introduced by Wojtas and Faliszewski [2012], is that voters or candidates have some probability of no-show. They investigated how to compute the winning probabilities of candidates, but not electoral control. They showed that for certain no-show probabilities, the problem of calculating winning probabilities can be reduced to a counting version of the electoral control problem, which is completely different from the CCAV-U problem because the latter involves arbitrary costs and arbitrary no-show probabilities of unregistered voters. As a matter of fact, the CCAV-U problem mandates new technical approaches.

From a technical point of view, the CCAV-U problem is related to stochastic combinatorial optimization, especially the stochastic knapsack problem (see Dean *et al.* [2008]; Bhalgat *et al.* [2011]). The prior work that is most closely related to ours is Kleinberg *et al.* [2000], which considered the stochastic knapsack problem with items of random weights but deterministic profits. Their goal is to find a subset of items so as to maximize the profit as long as the *overflow probability*, namely the probability that the total size exceeds a given knapsack, is no greater than a given parameter p. They provide an $O(\log p^{-1})$-approximation algorithm. As we will show in Section 2, the CCAV-U problem is equivalent to a different variant of the stochastic knapsack problem, namely the maximization of the overflow probability with respect to a determinisic constraint. It is not clear whether or not their result can be adapted to solve our problem, and whether or not our solution can be adapted to solve their problem.

Paper outline. The paper is organized as follows. Section 2 presents the model of the CCAV-U problem. Section 3 reviews two results that will be used in the paper. Section 4 presents the hardness result of the CCAV-U problem. Section 5 describes an approximation algorithm for the CCAV-U problem. Section 6 explores the issue of uncertainty in the broader context of electoral control. Section 7 concludes the paper with future research directions.

2 Problem Statement

Basic election model. In the basic election model, there are a set of candidates $\mathscr{C} = \{c_1, c_2, \ldots, c_r\}$ and a set of *registered* voters $\mathscr{V} = \{v_1, v_2, \ldots, v_s\}$. Each voter votes according to its preference of candidates. There is a voting rule according to which a winner is determined. In this paper we focus on the *plurality rule*, namely that every voter votes for its most

preferred candidate and the winner(s) will be the candidate(s) who receive(s) the highest number of votes.

The CCAV problem. In the classic CCAV (Constructive Control by Adding Voters) problem (see, e.g., Brandt *et al.* [2016]), there is an attacker who attempts to manipulate an election by adding a subset of *unregistered* voters, denoted by $\mathscr{W} = \{w_1, w_2, \cdots, w_n\}$ with $\mathscr{V} \cap \mathscr{W} = \emptyset$. The attacker is *constructive* in the sense that it attempts to make a designated candidate of its choice, say c_j^*, win the election. Since we focus on the plurality rule, the attacker only adds unregistered voters that will vote for c_{j^*}. However, adding an unregistered voter $w_i \in \mathscr{W}$ incurs a cost q_i to the attacker.

The CCAV-U problem. The CCAV-U problem extends the CCAV problem to accommodate the Uncertainty that the *unregistered* voters are not *reliable*, meaning that an unregistered voter may or may not cast its vote. In contrast, a registered voter is *reliable* in the sense that it *always* casts its vote (according to its own preference of candidates).

Specifically, there are a set of candidates $\mathscr{C} = \{c_1, c_2, \ldots, c_r\}$, a set of registered voters $\mathscr{V} = \{v_1, v_2, \ldots, v_s\}$, and a set of unregistered voters $\mathscr{W} = \{w_1, w_2, \cdots, w_n\}$. Suppose the attacker attempts to make a designated candidate of its choice, $c_{j^*} \in \mathscr{C}$, win the election, despite that c_{j^*} is not a winner with the registered voters only (i.e., when only the registered voters participate in the election). Suppose c_{j^*} can win the election after receiving at least additional k votes, which would come from the unregistered voters that are added by the attacker. Suppose adding unregistered voter $w_i \in \mathscr{W}$ incurs a cost of q_i to the attacker, but w_i will cast its vote only with probability p_i. The question is: given a fixed budget Q, subset $\mathscr{W}' \subseteq \mathscr{W}$ of unregistered voters should the attacker add in order to maximize the probability that c_{j^*} wins the election? More formally, we have:

The CCAV-U problem
Input: A set of candidates $\mathscr{C} = \{c_1, c_2, \ldots, c_r\}$; a set of registered voters $\mathscr{V} = \{v_1, v_2, \cdots, v_r\}$; a set of unregistered voters $\mathscr{W} = \{w_1, w_2, \cdots, w_n\}$, where w_i is associated with a cost q_i and a probability p_i; the plurality election rule; a designated candidate c_{j^*}; a parameter k (i.e., c_{j^*} needs additional k votes in order to win); attacker budget Q.
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 c_{j^*} wins with respect to voter set $\mathscr{V} \cup \mathscr{W}'$ is maximized, where $\mathscr{W}' = \{w_i | i \in I\}$.

The main difficulty of the CCAV-U problem is that when the attacker adds a subset $\mathscr{W}' \subseteq \mathscr{W}$ of unregistered voters, the subset $\tilde{\mathscr{W}}' \subseteq \mathscr{W}'$ of unregistered voters, who indeed cast their votes, is a *random* subset. That is, the realization of random subset $\tilde{\mathscr{W}}'$ will determine whether c_{j^*} wins or not. We remark that the cost q_i is incurred as long as w_i is added by the attacker, no matter whether w_i casts its vote or not. Therefore $\sum_{i \in I} q_i \leq Q$ is a deterministic constraint.

The KU problem. We observe that in order to solve the CCAV-U problem, it suffices to focus on the unregistered voters. This observation leads us to draw the following insight: the CCAV-U problem is equivalent to the following Knapsack with Uncertainty (KU) problem. Rather than introducing unnecessary notations, we reuse the notations n, q_i, p_i, and k of the CCAV-U problem in the KU problem because they respectively map to themselves between the two problems.

The KU problem
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 k.
Output: Find a set of indices $I \subseteq \{1, 2, \cdots, n\}$ such that

- $\sum_{i \in I} q_i \leq Q$, and
- $\Pr\left(\sum_{i \in I} P_i \geq k\right)$ is maximized.

Lemma 1. *The KU problem is equivalent to the CCAV-U problem.*

Proof. At a high level, each item in the KU problem can be seen as an unregistered voter in the CCAV-U problem and adding items can be seen as adding unregistered voters. Recall that c_{j^*} needs to receive at least k additional votes to win the election. This means that the attacker needs to add unregistered voters from which c_{j^*} receives at least additional k votes. Let $w_i \in W, i \in I$ be the subset of unregistered voters added by the attacker. Let $X_i \in \{0, 1\}$ be the random variable indicating whether w_i casts its vote or not, and let X_i have the same probability distribution as P_i in the KU problem, implying $\Pr(X_i = 1) = p_i$ and $\Pr(X_i = 0) = 1 - p_i$. The probability that c_{j^*} wins the election, namely that at least k unregistered voters cast their votes for c_{j^*}, is $\Pr\left(\sum_{i \in I} X_i \geq k\right) = \Pr\left(\sum_{i \in I} P_i \geq k\right)$. Hence, if there exists a subset of indices I such that $\sum_{i \in I} q_i \leq Q$ and the winning probability $\Pr\left(\sum_{i \in I} X_i \geq k\right)$ is maximized, then $\Pr\left(\sum_{i \in I} P_i \geq k\right)$ in the KU problem is maximized, and vice versa. □

Further notations for the KU problem. Let T be the set of items selected by the optimal solution to the KU problem, $\mathrm{OPT} = \Pr\left(\sum_{i \in T} P_i \geq k\right)$ be the optimal *objective value*. Abusing notations somewhat, we may also represent an item using its index, therefore T can also represent the set of the indices of the items corresponding to T. For any subset I of indices, let $P_I = \sum_{i \in I} P_i$ and $Q_I = \sum_{i \in I} q_i$.

3 Preliminaries

We will use the following inequalities to prove our results.

Markov's inequality (Stein and Shakarchi [2009]). Let Z be a random variable taking non-negative values. For any $a > 0$, it holds that

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

Berry-Essen theorem (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, $F_n(x)$ and $\Phi(x)$ are the cumulative distribution function of S_n and the standard normal distribution $\mathcal{N}(0,1)$, respectively, and

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

4 Inapproximability of the CCAV-U Problem

Summary of results. In this section we prove the inapproximability of the CCAV-U problem via the inapproximability of the KU problem.

Theorem 1. *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 k.*

Theorem 1 leads to:

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

Since the KU problem is equivalent to the CCAV-U problem, Theorem 1 and Corollary 1 lead to:

Corollary 2. *Assuming $P \neq NP$, there does not exist an $O(1)$-approximation algorithm for the CCAV-U problem that runs in polynomial time if k is part of the input. Moreover, assuming $W[1] \neq FPT$, there does not exist an $O(1)$-approximation algorithm for CCAV-U problem that runs in FPT time parameterized by k.*

Proof of Theorem 1. The proof strategy is to reduce the d-sum problem to the KU problem, while noting that the d-sum problem is $W[1]$-hard according to Downey and Fellows [1992]. We first review the d-sum problem.

Definition 1 (The d-sum problem)**.** *Given m positive integer $x_1, x_2, \cdots, x_m$ and an integer t, decide whether or not there exists a subset $E \subseteq \{x_1, x_2, \cdots, x_m\}$ such that $|E| = d$ and $\sum_{i:x_i \in E} x_i = t$.*

Proof of Theorem 1. At a high level, the proof is to show the following: If there is an α-approximation algorithm that solves the KU problem in $f(k)n^{O(1)}$ 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 of the d-sum problem.

Given an instance of the d-sum problem of m integers $x_1, x_2, \cdots, x_m$, we construct an instance of the KU problem as follows. Let $n = m$, $k = d$, and $\omega = \lceil \log_2 \alpha \rceil + 1$. We construct m items such that $p_i = 2^{-\omega x_i}$ and $q_i = M - \omega x_i$, where $M = m\omega \sum_{i=1}^{m} x_i$ and $i = 1, \ldots, m$. Let $Q = dM - \omega t$.

We make the following claims:

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

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

Given the two claims (as proven below), we show that any α-approximation algorithm for the KU problem can be used to solve the d-sum problem as follows. If the α-approximation algorithm returns a feasible solution with an objective value smaller than or equal to $2^{-\omega(t+1)}$, then the optimal objective value is at most $\alpha \cdot 2^{-\omega(t+1)} < 2^{-\omega t}$. In this case, claim (i) implies that the d-sum instance does not admit a feasible solution. If the α-approximation algorithm returns a feasible solution with an objective value larger than $2^{-\omega(t+1)}$, then claim (ii) implies that the d-sum instance must admit a feasible solution. Hence, any α-approximation algorithm for the KU problem can be used to solve the d-sum problem, and the theorem follows.

Now we prove the claims. For proving claim (i), suppose the d-sum problem admits a feasible solution E. Let $I = \{i | x_i \in E\}$ be the index set of items in the solution. We observe that

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

and

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

Hence, there exists a feasible solution with an objective value at least $2^{-\omega t}$. Claim (i) follows.

For proving claim (ii), suppose the d-sum problem does not admit a feasible solution.

Note that for any solution I to the KU problem, we have $|I| \leq d$ because $|I| \geq d+1$ 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 feasible solution. There are two possibilities: $|I| \leq d-1$ or $|I| = d$.

If $|I| \leq d-1$, then $\Pr(\sum_{i\in I} P_i \geq d) = 0 < 2^{-\omega(t+1)}$ and claim (ii) holds.

If $|I| = d$, then the fact that $\sum_{i\in I} q_i \leq Q$, $q_i = M - \omega x_i$ and $Q = dM - \omega t$ imply $\sum_{i\in I} x_i \geq t$. Since the d-sum problem does not admit a feasible solution, $|I| = d$ implies that either $\sum_{i:x_i\in I} x_i \geq t+1$ or $\sum_{i:x_i\in I} x_i \leq t-1$ holds. Given that $\sum_{i\in I} x_i \geq t$, we have $\sum_{i\in I} x_i \geq t+1$. This means

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

and claim (ii) holds. □

5 An Approximation Algorithm in FPT time

Summary of results. In this section we present an approximation algorithm for the KU , and therefore the CCAV-U , problem. The algorithm runs in FPT time for any fixed, small constant ε. This algorithm may *not* be seen as an *FPT scheme* in the traditional sense (see, e.g., Definition 7.3 in Chen [2016]). In terms of approximation ratio, our algorithm returns a value that has an additive error at most ε. Note that for classic combinatorial optimization problems,

an additive $O(1)$-approximation algorithm is usually considered to be better than a multiplicative one. However, for the KU and CCAV-U problems, we have $\mathrm{OPT} \in [0,1]$, meaning that an approximation solution with an additive $O(\varepsilon)$ error is not as good as an approximation algorithm with a multiplicative $O(\varepsilon)$ error. Nevertheless, the hardness result given by Theorem 1 suggests that additive approximation algorithm is perhaps the best we can hope for. On the other hand, it is not clear to us whether or not the running time can be further reduced to $f(k,\varepsilon)n^{O(1)}$ for some computable function f.

Theorem 2. *For any small constant $\varepsilon > 0$, there exists an algorithm for the* KU *problem which runs in $k^{O(k/\varepsilon)} + n^{O(1/\varepsilon^5)}$ time and returns a solution with an objective value no smaller than $\mathrm{OPT} - \varepsilon$, where $\mathrm{OPT} \in [0,1]$ is the optimal objective value in the* KU *problem.*

Since the KU problem is equivalent to the CCAV-U problem, we have the following corollary.

Corollary 3. *For any small constant $\varepsilon > 0$, there exists an algorithm for the* CCAV-U *problem which runs in $k^{O(k/\varepsilon)} + n^{O(1/\varepsilon^5)}$ time and returns a solution with an objective value no smaller than $\mathrm{OPT} - \varepsilon$, where $\mathrm{OPT} \in [0,1]$ is the optimal objective value in the* CCAV-U *problem.*

Proof strategy of Theorem 2. The proof is divided into three steps. The first step (Subsection 5.1) identifies the big items and the small items, and then prove Lemma 3 which shows the following: If we can find two approximation solutions whose objective values respectively almost match the contributions of the big items and the small items in the optimal solution, then one can combine these two approximation solutions to obtain a good approximation solution for the original KU problem. That is, Lemma 3 allows us to deal with big and small items separately.

The second step (Subsection 5.2) deals with big items. First we can assume the optimal solution selects at most $2k$ big items; otherwise, we can show (in Lemma 3) that the cheapest $2k$ big items already form a near-optimal solution. Given that at most $2k$ big items are selected by the optimal solution, we can round the probability of big items such that there are only $O(k/\varepsilon)$ distinct probabilities. This means that we can guess the number of big items corresponding to each rounded probability in the optimal solution, through which can select a proper subset of big items.

The third step (Subsection 5.3) handles small items. The basic idea is the following. When Berry-Essen's theorem is applicable, we 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 moments of a random variable are deterministic, we can use the techniques for solving the classical knapsack problem (Lemma 9). However, Berry-Essen's theorem is not always applicable. When Berry-Essen's theorem is not applicable, we are able to present a dynamic programming algorithm (Lemma 6) by utilizing Markov's inequality.

5.1 Identifying big and small items

For the KU problem, recall that k is a given parameter, P_i is a binary random variable where $\Pr(P_i = 1) = p_i$, T is the set of items selected by the optimal solution, and $P_I = \sum_{i \in I} P_i$ and $Q_I = \sum_{i \in I} q_i$ for any subset I of item indices.

Let $\varepsilon > 0$ be a fixed, small constant such that $1/\varepsilon \geq 4$ is an integer. We say an item in the KU problem is *big* if $p_i > 1 - \varepsilon^2$ and is *small* otherwise. Let S be the set of small items and B be the set of big items. Throughout this section, the notation A is always used to represent the set of items selected by our algorithm. Because our algorithm may select different sets of items in different circumstances, A may correspond to different sets of items in the proofs of different lemmas. Since we will deal with big and small items separately, $A \cap B$ and $A \cap S$ will be used to represent the set of *big* items and the set of *small* items that are selected by our algorithm, respectively.

The following lemma is a folklore.

Lemma 2. *Let Y_1, Y_2, Z_1, Z_2 be independent random variables that take values in $\mathbb{Z}_{\geq 0}$ (i.e., non-negative integers) such that for any integer $0 \leq h \leq N$ the following hold:*

$$\Pr(Y_1 \geq h) \geq (1-\delta)\Pr(Z_1 \geq h),$$
$$\Pr(Y_2 \geq h) \geq (1-\delta)\Pr(Z_2 \geq h).$$

Then, for any $0 \leq \ell \leq N$, the following holds:

$$\Pr(Y_1 + Y_2 \geq \ell) \geq (1-\delta)^2 \Pr(Z_1 + Z_2 \geq \ell).$$

Proof. For any integer $0 \leq \ell \leq N$, we have

$$\begin{aligned}
& \Pr(Y_1 + Y_2 \geq \ell) \\
= \; & \sum_{j=0}^{\ell-1} \Pr(Y_1 = j)\Pr(Y_2 \geq \ell - j) + \Pr(Y_1 \geq \ell) \\
\geq \; & (1-\delta)[\sum_{j=0}^{\ell-1} \Pr(Y_1 = j)\Pr(Z_2 \geq \ell - j) + \Pr(Y_1 \geq \ell)] \\
\geq \; & (1-\delta)\Pr(Y_1 + Z_2 \geq \ell).
\end{aligned}$$

Similarly, we can prove that

$$\Pr(Y_1 + Z_2 \geq \ell) \geq (1-\delta)\Pr(Z_1 + Z_2 \geq \ell).$$

Hence, $\Pr(Y_1 + Y_2 \geq \ell) \geq (1-\delta)^2 \Pr(Z_1 + Z_2 \geq \ell)$. □

Lemma 2 can be written additively as follows.

Corollary 4. *Let Y_1, Y_2, Z_1, Z_2 be independent random variables that take values in $\mathbb{Z}_{\geq 0}$ such that for any integer $0 \leq h \leq \ell$, the following hold:*

$$\Pr(Y_1 \geq h) \geq \Pr(Z_1 \geq h) - \delta,$$
$$\Pr(Y_2 \geq h) \geq \Pr(Z_2 \geq h) - \delta.$$

Then, the following holds:

$$\Pr(Y_1 + Y_2 \geq \ell) \geq \Pr(Z_1 + Z_2 \geq \ell) - 2\delta.$$

For an arbitrary solution $A \subseteq \{1, 2, \ldots, n\}$ to the KU problem, we can set $Y_1 = P_{A \cap S}$, $Y_2 = P_{A \cap B}$, $Z_1 = P_{T \cap S}$, and $Z_2 = P_{T \cap B}$, Combining Lemma 2 and Corollary 4, we know that in order to assure

$$\Pr(P_{A\cap S} + P_{A\cap B} \geq k) \geq (1-\delta)^2 \Pr(P_{T\cap S} + P_{T\cap B} \geq k) - 2\delta,$$

it suffices to assure the following for any integer $0 \leq h \leq k$:

$$\Pr(P_{A\cap S} \geq h) \geq (1-\delta)\Pr(P_{T\cap S} \geq h) - \delta, \quad (3a)$$
$$\Pr(P_{A\cap B} \geq h) \geq (1-\delta)\Pr(P_{T\cap B} \geq h) - \delta. \quad (3b)$$

We observe that a near-optimal solution can be found by setting $\delta = \Omega(\varepsilon)$. This observation leads us to investigate algorithms for coping with big items and small items separately.

5.2 Dealing with big items

Consider the number of big items selected by the optimal solution, namely $|T \cap B|$. We have either $|T \cap B| \geq 2k$ or $|T \cap B| < 2k$. In the case $|T \cap B| \geq 2k$, Lemma 3 below provides a polynomial-time algorithm that finds a solution that only consists of big items with an objective value at least $1-\varepsilon$, while ignoring the small items. In the case $|T \cap B| < 2k$, Lemma 4 below provides an FPT algorithm that returns a solution $A \cap B$ satisfying inequality Eq. (3b).

Lemma 3. *If $|T \cap B| \geq 2k$, then there is a polynomial-time algorithm that finds a solution A to the* KU *problem such that*

$$\Pr(P_A \geq k) \geq 1-\varepsilon \text{ and } Q_A \leq Q_{T\cap B} \leq Q.$$

Proof. Let A be the set of $2k$ big items with smallest sizes (among the big items). Since $|T \cap B| \geq 2k$, we have $Q_A \leq Q_{T\cap B} \leq Q$. Let $X_i = 1 - P_i$ and $\mu = \mathbb{E}(\sum_{i\in A} X_i) \leq 2k\varepsilon^2$. By applying Markov's inequality Eq. (1), we have

$$\begin{aligned}\Pr(P_A < k) &= \Pr\left(\sum_{i\in A} X_i \geq k+1\right)\\ &\leq \Pr\left(\sum_{i\in A} X_i \geq \mu \cdot \frac{1}{2\varepsilon^2}\right)\\ &\leq 2\varepsilon^2.\end{aligned}$$

Hence, $\Pr(P_A \geq k) \geq 1-\varepsilon$. □

We remark again that in case of $|T \cap B| \geq 2k$ we do not need to consider small items any more. In the following we assume $|T \cap B| < 2k$, whereas our algorithm deals with big and small items separately by returning $A \cap B$ and $A \cap S$.

Lemma 4. *If $|T \cap B| < 2k$, then there is an algorithm that runs in $k^{O(k/\varepsilon)}$ time and returns a set $A \cap B$ of big items such that*

- $Q_{A\cap B} \leq Q_{T\cap B}$,
- $\Pr(P_{A\cap B} \geq h) \geq (1-2\varepsilon)\Pr(P_{T\cap B} \geq h)$ *for any* $h \geq 0$.

Proof. We round the probabilities associated to big items as follows. Let $\delta = \varepsilon/k$ and $\gamma = O(1/\delta) = O(k/\varepsilon)$ be the largest integer such that $(1-\varepsilon^2)(1+\delta)^\gamma < 1$. Let

$$\Gamma_1 = \{1-\varepsilon^2, (1-\varepsilon^2)(1+\delta), \ldots, (1-\varepsilon^2)(1+\delta)^\gamma\}$$

be the set of rounded probabilities. For each big item, we round its probability p_i down to the nearest value in Γ_1 and denote it by $\tilde{p}_i$. Note that $\tilde{p}_i \leq p_i < \tilde{p}_i(1+\delta)$. Let B_j be the set of big items such that their associated probabilities are rounded to $(1-\varepsilon^2)(1+\delta)^j$.

For each $j \leq O(k/\varepsilon)$, we guess the value of $|T \cap B_j| \leq O(k)$. There are at most $k^{O(k/\varepsilon)}$ different possibilities on these values. Once we guess $|T \cap B_j|$ correctly for each j, we select the $|T \cap B_j|$ items that have the smallest size in B_j and let $\bar{B}_j$ denote the set of these items. Recall that $A \cap B$ represents the set of big items that are selected by our algorithm. Therefore, we can define $A \cap B = \cup_{j=0}^{\gamma} \bar{B}_j$ and claim that $A \cap B$ satisfies the condition required by Lemma 4. To see this, we observe that $|A \cap B_j| = |T \cap B_j|$ and $A \cap B$ consists of the items with the smallest size in B_j, therefore we have $Q_{A\cap B_j} \leq Q_{T\cap B_j}$ and consequently $Q_{A\cap B} \leq Q_{T\cap B}$. Now, we compare $\Pr(P_{A\cap B} = h)$ and $\Pr(P_{T\cap B} = h)$ for every $h \geq 0$. Let ϕ be an arbitrary one-to-one mapping that maps each item in $T \cap B_j$ to a distinct item in $A \cap B_j$ for every j. Then, we have

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

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

In order to show $\Pr(P_{A\cap B} = h) \geq (1-2\varepsilon)\Pr(P_{T\cap B} = h)$, it suffices to show that

$$\prod_{i\in I} p_{\phi(i)} \prod_{i\notin I}(1-p_{\phi(i)}) \geq (1-2\varepsilon)\prod_{i\in I} p_i \prod_{i\notin I}(1-p_i)$$

for every $I \subseteq T \cap B$ with $|I| = h$. According to the way we round probabilities, we have $p_{\phi(i)} \leq p_i < p_{\phi(i)}(1+\delta)$, hence $1-p_{\phi(i)} \geq 1-p_i$ and

$$\prod_{i\in I} p_{\phi(i)} \geq (1-\delta)^h \prod_{i\in I} p_i \geq (1-h\delta)\prod_{i\in I} p_i.$$

Since $h \leq 2k$, we have $h\delta \leq 2\varepsilon$ and

$$\Pr(P_{A\cap B} = h) \geq (1-2\varepsilon)\Pr(P_{T\cap B} = h)$$

for any $h \geq 0$. Therefore, we have

$$\Pr(P_{A\cap B} \geq h) \geq (1-2\varepsilon)\Pr(P_{T\cap B} \geq h).$$

The lemma follows. □

5.3 Dealing with small items

Recall that for a small item i we have $\varepsilon^2 \leq 1-p_i \leq 1$. By re-indexing the items, we can assume without loss of generality that $S = \{1, 2, \cdots, n'\}$. Note that $n' \leq n$. Our goal is to prove the following lemma.

Lemma 5. *There exists an algorithm that runs in $n^{O(1/\varepsilon^5)}$ time and returns a feasible solution $A \cap S$ such that*

$$Q_{A\cap S} \leq Q_{T\cap S} \text{ and } \Pr(P_{A\cap S} \geq h) \geq \Pr(P_{T\cap S} \geq h) - \Omega(\varepsilon)$$

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

For proving Lemma 5, we consider the following two cases separately: $\sum_{i\in T\cap S} p_i \leq (1/\varepsilon)^4$ and $\sum_{i\in T\cap S} p_i > (1/\varepsilon)^4$.

Case 1: $\sum_{i\in T\cap S} p_i \leq (1/\varepsilon)^4$.
By Markov's inequality Eq. (1), we know that $\Pr(P_{T\cap S} \geq h) \leq \varepsilon$ for $h \geq (1/\varepsilon)^5$. Let $\zeta = (1/\varepsilon)^5$. Lemma 6 below says that we can find a subset $A \cap S$ of items in polynomial time such that

$$\Pr(P_{A\cap S} = h) \leq \Pr(P_{T\cap S} = h) + 2\varepsilon/n$$

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

$$\Pr(P_{A\cap S} \geq h) \geq \Pr(P_{T\cap S} \geq h) - 2\varepsilon$$

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

$$\Pr(P_{A\cap S} \geq h) \geq 0 \geq \Pr(P_{T\cap S} \geq h) - 2\varepsilon$$

for $h \geq \zeta$, we find a near-optimal solution $A \cap S$ in polynomial time.

What remains to be done is to prove Lemma 6 below.

Lemma 6. *For any $\zeta \le (1/\varepsilon)^{O(1)}$, there exists an algorithm that runs in $n^{(1/\varepsilon)^{O(1)}}$ time and returns a solution $A \cap S$ such that $Q_{A\cap S} \le Q_{T\cap S}$ and $\Pr(P_{A\cap S} = h) \le \Pr(P_{T\cap S} = h) + 2\varepsilon/n$ for every $0 \le h \le \zeta - 1$.*

Proof. We design an algorithm based on dynamic programming. Let $\eta = \varepsilon/n^2$. Although we do not know the value of $\Pr(P_{T\cap S} = h)$, we know that this value lies in $[0,1]$. Therefore, we can guess, via $(n/\varepsilon)^{O(\zeta)} = n^{(1/\varepsilon)^{O(1)}}$ enumerations, the ζ values $t_0, t_1, \cdots, t_{\zeta-1}$ such that $t_h - \varepsilon/n \le \Pr(P_{T\cap S} = h) < t_h$. In the following we provide an algorithm that returns $A \cap S$ such that $\Pr(P_{A\cap S} = h) \le t_h + \varepsilon/n$, and Lemma 6 follows.

Let us define $\Gamma_2 = \{0, \eta, 2\eta, \ldots, \eta \cdot 1/\eta\}$ as the set of rounded probabilities. Let us call a $(\zeta+1)$-vector $(j, u_0, u_1, u_2, \ldots, u_{\zeta-1})$ a *state*, where $j \in \{0, 1, \ldots, n'\}$ and $u_j \in \Gamma_2$. Each state is associated with a positive value $f(j, u_0, u_1, \ldots, u_{\zeta-1})$, which can be calculated recursively as shown in the next paragraph. Intuitively, a state means that a subset $U \subseteq \{1, 2, \ldots, j\}$ of items can be selected such that $\Pr(P_U = j)$ is approximately u_j, and $f(j, u_0, u_1, \ldots, u_{\zeta-1})$ is the minimal total size of items among all possible subsets U. In particular, if such a subset U does not exist, then $f(j, u_0, u_1, \ldots, u_{\zeta-1}) = \infty$.

Now we define the calculation of $f(j, u_0, u_1, \ldots, u_{\zeta-1})$. For this purpose, we first define the summation of state $(j, u_0, u_1, \ldots, u_{\zeta-1})$ and random variable P_{j+1} as follows:

$$(j, u_0, u_1, \ldots, u_{\zeta-1}) + P_{j+1} = (j+1, \tilde{u}'_0, \tilde{u}'_1, \ldots, \tilde{u}'_{\zeta-1}), \quad (4)$$

where $\tilde{u}'_j$ is the nearest value in Γ_2 when rounding *up* u'_j with $u'_0 = u_0(1 - p_{j+1})$ and $u'_j = u_j(1 - p_{h+1}) + u_{j-1} p_h$ for $1 \le j \le \zeta - 1$.

Initially, for $j = 0$, we define

$$f(0, 0, 0, \ldots, 0) = 0 \quad (5)$$

and for $(u_0, u_1, \ldots, u_{\zeta-1}) \ne (0, 0, \ldots, 0)$, we define

$$f(0, u_0, u_1, \ldots, u_{\zeta-1}) = \infty. \quad (6)$$

For $h \ge 0$, we define

$$g(j+1, u_0, u_1, \ldots, u_{\zeta-1}) = q_{j+1} + \min\{f(j, u'_0, u'_1, \ldots, u'_{\zeta-1}) : (j, u'_0, u'_1, \ldots, u'_{\zeta-1}) + P_{j+1} = (j+1, u_0, u_1, \ldots, u_{\zeta-1})\},$$

and define

$$\begin{aligned} & f(j+1, u_0, u_1, \ldots, u_{\zeta-1}) \\ = \ & \min\{f(j, u_0, u_1, \ldots, u_{\zeta-1}), g(j+1, u_0, u_1, \ldots, u_{\zeta-1})\}. \end{aligned} \quad (7)$$

Observe that we can use Eqs. (5)-(7) to recursively calculate the value associated to any state. Since the total number of states is bounded from above by $|\Gamma_2|^{O(\zeta)} = (n/\varepsilon)^{(1/\varepsilon)^{O(1)}}$, the calculation can be done in polynomial time. In the following we show that, among all of the states $(n, u_0, u_1, \ldots, u_{\zeta-1})$ that satisfy $f(n, u_0, u_1, \ldots, u_{\zeta-1}) \le Q$, there exists some state $(n, u^*_0, u^*_1, \ldots, u^*_{\zeta-1})$ such that $u^*_h \le t_h + \varepsilon/n$. Denote the set of items selected in the corresponding solution by $A \cap S$, then $A \cap S$ satisfies Lemma 6.

Consider the optimal solution $T \cap S$. Let $T_j = T \cap S \cap \{1, 2, \ldots, j\}$ and $u_i(T_j) = \Pr(P_{T_j} = i)$. We make the following claim.

Claim 1. *For any $0 \le j \le n'$, there exists a vector $(j, u_0, u_1, \ldots, u_{\zeta-1})$ such that*

- *$f(j, u_0, u_1, \ldots, u_{\zeta-1}) \le Q_{T_j}$, and*
- *$u_i \le u_i(T_j) + j\eta$ for every $0 \le i \le \zeta - 1$.*

Proof of Claim 1. We prove the claim by induction. It is trivial to see that the claim holds when $j = 0$. Suppose the claim holds for $j \le \ell$. That is, for $j = \ell$, there exists some state $(\ell, u_0, u_1, \ldots, u_{\zeta-1})$ such that $u_i \le u_i(T_\ell) + \ell\eta$ for every $0 \le i \le \zeta - 1$ and $f(\ell, u_0, u_1, \ldots, u_{\zeta-1}) \le Q_{T_\ell}$.

Now we prove it holds for $j = \ell + 1$. There are two cases: $\ell + 1 \notin T_{\ell+1}$ and $\ell + 1 \in T_{\ell+1}$.

In the case $\ell + 1 \notin T_{\ell+1}$, we have $Q_{T_{\ell+1}} = Q_{T_\ell}$ and $u_i(T_{\ell+1}) = u_i(T_\ell)$. According to Equation (7), we have

$$f(\ell+1, u_0, u_1, \ldots, u_{\zeta-1}) \le f(\ell, u_0, u_1, \ldots, u_{\zeta-1}) \le Q_{T_\ell} = Q_{T_{\ell+1}},$$

hence the claim holds.

In the case $\ell + 1 \in T_{\ell+1}$, we have

$$u_0(T_{\ell+1}) = u_0(T_\ell) p_{\ell+1}, \quad (8)$$

$$u_i(T_{\ell+1}) = u_i(T_\ell)(1 - p_{\ell+1}) + u_{i-1}(T_\ell) p_{\ell+1}, 1 \le i \le \zeta - 1 \quad (9)$$

Compare Eqs. (8)-(9) with (7), we see that if

$$(\ell+1, u'_0, u'_1, \ldots, u'_{\zeta-1}) = (\ell, u_0, u_1, \ldots, u_{\zeta-1}) + P_{\ell+1},$$

then

$$u'_0 \le u_0(1 - p_{\ell+1}) + \eta \le u_0(T_{\ell+1}) + (\ell+1)\eta,$$

and

$$\begin{aligned} u'_i &\le u_i(1 - p_{\ell+1}) + u_{i-1} p_{\ell+1} + \eta \\ &\le u_i(T_\ell)(1 - p_{\ell+1}) + u_{i-1}(T_\ell) p_{\ell+1} + (\ell+1)\eta \\ &= u_i(T_{\ell+1}) + (\ell+1)\eta. \end{aligned}$$

Furthermore, we have

$$f(\ell+1, u'_0, u'_1, \ldots, u'_{\zeta-1}) \le f(\ell, u_0, u_1, \ldots, u_{\zeta-1}) + q_{\ell+1} \le Q_{T_{\ell+1}}.$$

Hence, the claim also holds. □

Now we can finish the proof of the lemma. Claim 1 says that there exists a state $(n, u^*_0, u^*_1, \ldots, u^*_{\zeta-1})$ such that $f(n, u^*_0, u^*_1, \ldots, u^*_{\zeta-1}) \le Q$ and $u^*_i \le u_i(T_{n'}) + n'\eta \le u_i(T \cap S) + \varepsilon/n$. Since in the recursive calculation we always overestimate (by rounding *up*) the probabilities, we have

$$\Pr(P_{A\cap S} = h) \le u^*_h \le \Pr(P_{T\cap S} = h) + \varepsilon/n \le t_h + \varepsilon/n.$$

Hence, Lemma 6 follows. □

Case 2: $\sum_{i\in T\cap S} p_i > (1/\varepsilon)^4$.
For any subset D of small items and integer $h \ge 0$, we define

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

Then the following is true.

Lemma 7. *If $\sum_{i\in A\cap S} p_i > (1/\varepsilon)^4$ and $|\Phi(\hat{h}_{A\cap S}) - \Phi(\hat{h}_{T\cap S})| \le O(\varepsilon)$, then $\Pr(\sum_{i\in A\cap S} P_i \ge h) \ge \Pr(\sum_{i\in T\cap S} P_i \ge h) - \Omega(\varepsilon)$, where $\Phi(x)$ is the cumulative distribution function of the standard normal distribution.*

Proof. We define random variable $X_i = P_i - \mathbb{E}(P_i) = P_i - p_i$, then $\mathbb{E}(X_i) = 0$,

$$\sigma_i^2 = \mathbb{E}(X_i^2) = (1-p_i)^2 p_i + p_i^2(1-p_i) = p_i(1-p_i),$$
$$\rho_i = \mathbb{E}(|X_i|^3) = p_i(1-p_i)[p_i^2 + (1-p_i)^2].$$

We have

$$\Pr\left(\sum_{i\in A\cap S} P_i \geq h\right) \quad = \quad \Pr\left(\frac{\sum_{i\in A\cap S} X_i}{\sqrt{\sum_{i\in A\cap S}\sigma_i^2}} \geq \hat{h}_{A\cap S}\right).$$

According to Berry-Essen's theorem (2), we have

$$\left|\Pr\left(\sum_{i\in A\cap S} P_i \geq h\right) - \left(1-\Phi(\hat{h}_{A\cap S})\right)\right| \leq C_0 \cdot \frac{\sum_{i\in A\cap S}\rho_i}{(\sum_{i\in A\cap S}\sigma_i^2)^{3/2}}.$$

By plugging in ρ_i and σ_i, we have

$$\left|\Pr\left(\sum_{i\in A\cap S} P_i \geq h\right) - \left(1-\Phi(\hat{h}_{A\cap S})\right)\right| \leq C_0 \cdot \frac{1}{\sqrt{\sum_{i\in A\cap S} p_i(1-p_i)}}.$$

For small items, it holds that $1-p_i \geq \varepsilon^2$. Since $\sum_{i\in A\cap S} p_i > (1/\varepsilon)^4$, we have

$$\left|\Pr\left(\sum_{i\in A\cap S} P_i \geq h\right) - \left(1-\Phi(\hat{h}_{A\cap S})\right)\right| \leq C_0\varepsilon.$$

Hence, the probability $\Pr(\sum_{i\in A\cap S} P_i \geq h)$ can be estimated using the standard normal distribution $\Phi(\hat{h}_{A\cap S})$. More specifically, if we can select $A\cap S$ such that

$$|\Phi(\hat{h}_{A\cap S}) - \Phi(\hat{h}_{T\cap S})| \leq O(\varepsilon)$$

for every $0 \leq h \leq k$, then it holds that

$$\begin{aligned}\Pr\left(\sum_{i\in A\cap S} P_i \geq h\right) &\geq \left(1-\Phi(\hat{h}_{A\cap S})\right) - C_0\varepsilon \\ &= \left(1-\Phi(\hat{h}_{T\cap S})\right) - \Omega(\varepsilon) \\ &\geq \Pr\left(\sum_{i\in T\cap S} P_i \geq h\right) - \Omega(\varepsilon). \quad \square\end{aligned}$$

The following lemma allows us to transform the condition of $|\Phi(\hat{h}_{A\cap S}) - \Phi(\hat{h}_{T\cap S})| \leq O(\varepsilon)$ to an even more straightforward one.

Lemma 8. *If $\sum_{i\in T\cap S} p_i > (1/\varepsilon)^4$ and the following hold for some $A\cap S$:*

- $|\sum_{i\in A\cap S} p_i - \sum_{i\in T\cap S} p_i| \leq O(\varepsilon)$, *and*
- $|\sum_{i\in A\cap S} p_i(1-p_i) - \sum_{i\in T\cap S} p_i(1-p_i)| \leq O(\varepsilon)$,

then $|\Phi(\hat{h}_{A\cap S}) - \Phi(\hat{h}_{T\cap S})| \leq O(\varepsilon)$ for every $0 \leq h \leq k$.

Towards the proof, we need the following claim.

Claim 2. *For any $x \in (-\infty,\infty)$ and $\delta > 0$, it holds that $|\Phi((1+\delta)x \pm \delta) - \Phi(x)| \leq 2\delta$.*

Proof. We observe that for any $y \in (-\infty,\infty)$, it holds that

$$|\Phi(y\pm\delta) - \Phi(y)| = \left|\int_y^{y\pm\delta} \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt\right| \leq \left|\int_y^{y\pm\delta} 1\, dt\right| = \delta.$$

Now we show

$$|\Phi((1+\delta)x) - \Phi(x)| = \left|\int_x^{x+\delta x} \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt\right| \leq \delta.$$

We observe that because of the symmetry in $x \in (-\infty,\infty)$, it suffices to prove the above inequality for $x \geq 0$. In this case, we have $e^{-t^2/2} \leq 1/t$ for $t \geq 0$. (This is because the derivative of $te^{-t^2/2}$ is $e^{t^2/2}(1-t^2)$, and consequently its maximum value is $1/\sqrt{e} \leq 1$.) Therefore, we have

$$\left|\int_x^{x+\delta x} \frac{1}{\sqrt{2\pi}} e^{-t^2/2} dt\right| \leq \delta x \cdot 1/x = \delta$$

for $x \geq 0$. Hence, Claim 2 holds. $\square$

Now we can prove Lemma 8.

Proof of Lemma 8. Note that $|\sum_{i\in A\cap S} p_i(1-p_i) - \sum_{i\in T\cap S} p_i(1-p_i)| \leq \varepsilon$ and $\sum_{i\in T\cap S} p_i > (1/\varepsilon)^4$. We define

$$\begin{aligned}\beta_1 &= \sqrt{\frac{\sum_{i\in A\cap S} p_i(1-p_i)}{\sum_{i\in T\cap S} p_i(1-p_i)}} \in [1-\varepsilon, 1+\varepsilon], \\ \beta_2 &= \sum_{i\in A\cap S} p_i - \sum_{i\in T\cap S} p_i \in [-\varepsilon,\varepsilon].\end{aligned}$$

Then, we have

$$\begin{aligned}\hat{h}_{A\cap S} = \frac{h - \sum_{i\in A\cap S} p_i}{\sqrt{\sum_{i\in A\cap S} p_i(1-p_i)}} &= \frac{1}{\beta_1}\cdot\frac{h - \sum_{i\in T\cap S} p_i - \beta_2}{\sqrt{\sum_{i\in T\cap S} p_i(1-p_i)}} \\ &= \frac{1}{\beta_1}\cdot\hat{h}_{T\cap S} + O(\varepsilon)\end{aligned}$$

Claim 2 implies $|\Phi(\hat{h}_{A\cap S}) - \Phi(\hat{h}_{T\cap S})| \leq O(\varepsilon)$. $\square$

Now it suffices to show the following Lemma, which directly implies Lemma 5 for the case of $\sum_{i\in T\cap S} p_i > (1/\varepsilon)^4$.

Lemma 9. *If $\sum_{i\in T\cap S} p_i > (1/\varepsilon)^4$, then there exists an algorithm that runs in $O(n^4/\varepsilon^2)$ time and returns a feasible solution with item set $A\cap S$ such that $Q_{A\cap S} \leq Q_{T\cap S}$ and*

- $|\sum_{i\in A\cap S} p_i - \sum_{i\in T\cap S} p_i| \leq O(\varepsilon)$, *and*
- $|\sum_{i\in A\cap S} p_i(1-p_i) - \sum_{i\in T\cap S} p_i(1-p_i)| \leq O(\varepsilon)$.

Proof. We do not know the values of $\sum_{i\in T\cap S} p_i$ and $\sum_{i\in T\cap S} p_i(1-p_i)$. However, as they lie in $[1/\varepsilon^2, n]$, we can guess (via n^2/ε^2 enumerations) the two values that have an error no greater than ε. Denote these two values respectively by t_1 and t_2. In what follows we present a dynamic programming algorithm that returns a solution $A\cap S$ such that $|\sum_{i\in A\cap S} p_i - t_1| \leq O(\varepsilon)$, $|\sum_{i\in A\cap S} p_i(1-p_i) - t_2| \leq O(\varepsilon)$, and $Q_{A\cap S} \leq Q_{T\cap S}$. The algorithm uses the same idea as the one for solving the classical knapsack problem. For self-containedness, we describe the algorithm below.

Define $\eta' = \varepsilon/n$. Let $\Gamma_3 = \{\eta', 2\eta', \ldots, n^2/\varepsilon \cdot \eta'\}$ be the set of rounded values. Define (j, u_1, u_2) as a state, where $0 \leq j \leq |S| = n'$ and $u_1, u_2 \in \Gamma_3$. Each state is associated with a value $f(j, u_1, u_2)$, which can be calculated recursively as shown in the next paragraph. Intuitively, a state means that a subset $U \subseteq \{1, 2, \ldots, j\}$ can be selected such that $\sum_{i \in U} p_i$ and $\sum_{i \in U} p_i(1-p_i)$ are respectively approximately u_1 and u_2, and $f(u_1, u_2)$ is the minimal total size of items among all possible subsets U.

Now we define the value of $f(j, u_1, u_2)$. For this purpose, we first define the summation of state (j, u_1, u_2) and random variable P_{j+1} as follows:

$$(j, u_1, u_2) + P_{j+1} = (j+1, \tilde{u}'_1, \tilde{u}'_2),$$

where $\tilde{u}'_1$ and $\tilde{u}'_2$ are respectively the nearest value in Γ_3 when rounding up $u'_1 = u_1 + p_{j+1}$ and $u'_2 = u_2 + p_{j+1}(1-p_{j+1})$. For $j = 0$, we define $f(0,0,0) = 0$ and for $(u_1, u_2) \neq (0,0)$ we define $f(0, u_1, u_2) = \infty$. For $j \geq 0$, we define

$$g(j+1, u_1, u_2) = q_{j+1} + \min\{f(j, u'_1, u'_2) : (j, u'_1, u'_2) + P_{j+1} = (j+1, u_1, u_2)\}$$

and

$$f(j+1, u_1, u_2) = \min\{f(j, u_1, u_2), g(j+1, u_1, u_2)\} \quad (10)$$

The preceding definitions allow us to recursively calculate the value associated to any state. Since the total number of states is bounded from above by n^4/ε^2, the calculation can be done in polynomial time.

What remains to be done is to show that among all of the states $(|S|, u_1, u_2)$, there exists some $(|S|, u^*_1, u^*_2)$ such that $|u^*_1 - t_1| \leq \varepsilon$, $|u^*_2 - t_2| \leq \varepsilon$ and $f(|S|, u^*_1, u^*_2) \leq Q_{T \cap S}$. For this purpose, let us consider the optimal solution $T \cap S$. Let $T_j = T \cap S \cap \{1, 2, \ldots, j\}$ and $u_i(T_j) = \Pr(P_{T_j} = i)$ for $i = 1, 2$. We make the following claim.

Claim 3. *For any $j \geq 0$, there exists some (j, u_1, u_2) such that*

- *$f(j, u_1, u_2) \leq Q_{T_j}$, and*
- *$u_i \leq u_i(T_j) + j\eta'$ for $i = 1, 2$.*

Proof of Claim 3. We prove the claim by induction. The claim holds trivially for $j = 0$. Suppose the claim holds for $j \leq \ell$. That is, for $j = \ell$, there exists some state (ℓ, u_1, u_2) such that $f(\ell, u_1, u_2) \leq Q_{T_\ell}$ and $u_i \leq u_i(T_\ell) + \ell\eta'$ for $i = 1, 2$.

Now we prove that the claim holds for $j = \ell + 1$. There are two cases: $\ell + 1 \notin T_{\ell+1}$ and $\ell + 1 \in T_{\ell+1}$. In the case $\ell + 1 \notin T_{\ell+1}$, we have $Q_{T_{\ell+1}} = Q_{T_\ell}$ and $u_i(T_{\ell+1}) = u_i(T_\ell)$. According to Eq. (10), we have $f(\ell+1, u_1, u_2) \leq f(\ell, u_1, u_2) \leq Q_{T_\ell} = Q_{T_{\ell+1}}$, hence the claim. In the case $\ell + 1 \in T_{\ell+1}$, we have

$$u_1(T_{\ell+1}) = u_1(T_\ell) + p_{\ell+1}, \quad (11)$$

$$u_2(T_{\ell+1}) = u_2(T_\ell) + p_{\ell+1}(1-p_{\ell+1}). \quad (12)$$

Compare Eqs. (11)-(11) and (10), we see that if $(\ell+1, u'_1, u'_2) = (\ell, u_1, u_2) + P_{\ell+1}$, then

$$u'_1 \leq u_1 + p_{\ell+1} + \eta' \leq u_1(T_{\ell+1}) + (\ell+1)\eta',$$

$$u'_2 \leq u_2 + p_{\ell+1}(1-p_{\ell+1}) + \eta' \leq u_2(T_{\ell+1}) + (\ell+1)\eta',$$

and $f(\ell+1, u'_1, u'_2) \leq f(\ell, u_1, u_2) + q_{\ell+1} \leq Q_{T_{\ell+1}}$. Hence, the claim holds. □

Claim 3 says that there exists some state $(|S|, u_1, u_2)$ such that $f(|S|, u_1, u_2) \leq Q$ and $u_i \leq u_i(T) + n\eta' = u_i(T) + \varepsilon$, which completes the proof of Lemma 9. □

Given our discussion on the two cases, we know Lemma 5 is true, and consequently we have Theorem 2.

6 Discussion

Having investigated the hardness and algorithmic aspects of CCAV-U under the plurality voting rule, it is natural to ask whether or not our techniques can be used in other closely related problems. As shown in the following two examples of closely related problems, they however may demand different techniques. In particular, it is not clear how we can transform them into a stochastic knapsack problem.

One closely related problem is to investigate the CCAV-U problem under other voting rules, say, r-approval for $r \geq 2$. However, it is not clear whether or not our techniques can be applied to the case of $r = 2$. Note that for 2-approval, adding an unregistered voter may contribute to the votes of both the designated candidate and some of the other candidates. That is, even if the designated candidate needs k additional votes to win before adding any unregistered voter, it may not be sufficient to add unregistered voters to make the designated candidate get k additional votes. This is because some of the other candidates also get votes from these unregistered voters. This makes the problem more complicated.

Another closely related problem is to investigate whether or not our techniques can be applied to models of other kinds of attacks (e.g., destructive attacks) or other types of manipulation (e.g., deleting voters). In the case of manipulation by deleting voters, there is no concept of unregistered voters. By introducing uncertainty to all of the voters (i.e., probabilities of no-show), the number of votes received by each candidate is a random variable, which makes the problem harder.

7 Conclusion

We investigated the hardness and algorithmic aspects of CCAV-U under the plurality voting rule, which accommodates the uncertainty that the unregistered voters added by the attacker may or may not cast their votes. We showed that CCAV-U does not admit any *multiplicative* $O(1)$-approximation algorithm in FPT time (parameterized by k) modulo standard complexity assumptions. We also showed that there is an algorithm that returns an approximate solution with an *additive* ε-error in FPT time for any fixed ε. Given the hardness result, this algorithm is perhaps the best one can hope for.

This paper introduces a range of open problems. In addition to those mentioned in Section 6, it is interesting to investigate whether or not the CCAV-U problem can be solved in $f(\varepsilon, k)n^{O(1)}$ time with an *additive* ε-approximation solution, where f is a computable function.

References

John J Bartholdi, Craig A Tovey, and Michael A Trick. How hard is it to control an election? *Mathematical and Computer Modelling*, 16(8-9):27–40, 1992.

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

Dorothea Baumeister, Magnus Roos, and Jörg Rothe. Computational complexity of two variants of the possible winner problem. In *AAMAS*, pages 853–860. IFAAMAS, 2011.

Dorothea Baumeister, Magnus Roos, Jörg Rothe, Lena Schend, and Lirong Xia. The possible winner problem with uncertain weights. In *ECAI*, pages 133–138. IOS Press, 2012.

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

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

Nadja Betzler, Susanne Hemmann, and Rolf Niedermeier. A multivariate complexity analysis of determining possible winners given incomplete votes. In *IJCAI*, volume 9, pages 53–58, 2009.

Anand Bhalgat, Ashish Goel, and Sanjeev Khanna. Improved approximation results for stochastic knapsack problems. In *SODA*, pages 1647–1665. SIAM, 2011.

Craig Boutilier, Jérôme Lang, Joel Oren, and Héctor Palacios. Robust winners and winner determination policies under candidate uncertainty. In *AAAI*, pages 1391–1397, 2014.

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

Jiehua Chen. *Exploiting structure in computationally hard voting problems*, volume 6. Universitätsverlag der TU Berlin, 2016.

Yann Chevaleyre, Jérôme Lang, Nicolas Maudet, and Jérôme Monnot. Possible winners when new candidates are added: The case of scoring rules. In *AAAI*, 2010.

Brian C Dean, Michel X Goemans, and Jan Vondrák. Approximating the stochastic knapsack problem: The benefit of adaptivity. *Mathematics of Operations Research*, 33(4):945–964, 2008.

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

Gábor Erdélyi, Lena Piras, and Jörg Rothe. The complexity of voter partition in bucklin and fallback voting: Solving three open problems. In *AAMAS*, pages 837–844. IFAAMAS, 2011.

Piotr Faliszewski, Edith Hemaspaandra, Lane Hemaspaandra, and Jörg Rothe. A richer understanding of the complexity of election systems. *Fundamental problems in computing: Essays in honor of Professor Daniel J. Rosenkrantz*, pages 375–406, 2009.

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, and Lane A Hemaspaandra. Using complexity to protect elections. *Communications of the ACM*, 53(11):74–82, 2010.

Piotr Faliszewski, Edith Hemaspaandra, and Lane A Hemaspaandra. Weighted electoral control. In *AAMAS*, pages 367–374. IFAAMAS, 2013.

Zack Fitzsimmons, Edith Hemaspaandra, and Lane A Hemaspaandra. Control in the presence of manipulators: Cooperative and competitive cases. In *IJCAI*, pages 113–119, 2013.

Noam Hazon, Yonatan Aumann, Sarit Kraus, and Michael Wooldridge. On the evaluation of election outcomes under uncertainty. *Artificial Intelligence*, 189:1–18, 2012.

Edith Hemaspaandra, Lane A Hemaspaandra, and Jörg Rothe. Anyone but him: The complexity of precluding an alternative. *Artificial Intelligence*, 171(5-6):255–285, 2007.

Jon Kleinberg, Yuval Rabani, and Éva Tardos. Allocating bandwidth for bursty connections. *SIAM Journal on Computing*, 30(1):191–217, 2000.

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

Andrew Peter Lin. *Solving hard problems in election systems*. Rochester Institute of Technology, 2012.

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

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

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

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