Thompson Sampling for Real-Valued Combinatorial Pure Exploration of Multi-Armed Bandit

Shintaro Nakamura[1,2], **Masashi Sugiyama** [2, 1]
[1]The University of Tokyo
[2]RIKEN AIP
nakamurashintaro@g.ecc.u-tokyo.ac.jp, sugi@k.u-tokyo.ac.jp

Abstract

We study the real-valued combinatorial pure exploration of the multi-armed bandit (R-CPE-MAB) problem. In R-CPE-MAB, a player is given d stochastic arms, and the reward of each arm $s \in \{1, \ldots, d\}$ follows an unknown distribution with mean μ_s. In each time step, a player pulls a single arm and observes its reward. The player's goal is to identify the optimal *action* $\boldsymbol{\pi}^* = \underset{\boldsymbol{\pi} \in \mathcal{A}}{\arg\max}\, \boldsymbol{\mu}^\top \boldsymbol{\pi}$ from a finite-sized real-valued *action set* $\mathcal{A} \subset \mathbb{R}^d$ with as few arm pulls as possible. Previous methods in the R-CPE-MAB require enumerating all of the feasible actions of the combinatorial optimization problem one is considering. In general, since the size of the action set grows exponentially large in d, this is almost practically impossible when d is large. We introduce an algorithm named the Generalized Thompson Sampling Explore (GenTS-Explore) algorithm, which is the first algorithm that can work even when the size of the action set is exponentially large in d. We also introduce a novel problem-dependent sample complexity lower bound of the R-CPE-MAB problem, and show that the GenTS-Explore algorithm achieves the optimal sample complexity up to a problem-dependent constant factor.

Introduction

Pure exploration in the stochastic multi-armed bandit (PE-MAB) is one of the important frameworks for investigating online decision-making problems, where we try to identify the optimal object from a set of candidates as soon as possible (Bubeck, Munos, and Stoltz 2009; Audibert, Bubeck, and Munos 2010; Chen et al. 2014). One of the important models in PE-MAB is the *combinatorial pure exploration* task in the multi-armed bandit (CPE-MAB) problem (Chen et al. 2014; Wang and Zhu 2022; Gabillon et al. 2016; Chen, Gupta, and Li 2016; Chen et al. 2017). In CPE-MAB, we have a set of d stochastic arms, where the reward of each arm $s \in \{1, \ldots, d\}$ follows an unknown distribution with mean μ_s, and a finite-sized *action set* $\mathcal{A}$, which is a collection of subsets of arms with certain combinatorial structures. The size of the action set can be exponentially large in d. In each time step, a player pulls a single arm and observes a reward from it. The goal is to identify the best action from action set $\mathcal{A}$ with as few arm pulls as possible. Abstractly, the goal is to identify $\boldsymbol{\pi}^*$, which is the optimal solution for the following constraint optimization problem:

$$\begin{array}{ll} \text{maximize}_{\boldsymbol{\pi}} & \boldsymbol{\mu}^\top \boldsymbol{\pi} \\ \text{subject to} & \boldsymbol{\pi} \in \mathcal{A}, \end{array} \tag{1}$$

where $\boldsymbol{\mu}$ is a vector whose s-th element is the mean reward of arm s and $\top$ denotes the transpose. One example of the CPE-MAB is the shortest path problem shown in Figure 1. Each edge $s \in \{1, \ldots, 7\}$ has a cost μ_s and $\mathcal{A} = \{(1,0,1,0,0,1,0), (0,1,0,1,0,1,0), (0,1,0,0,1,0,1)\}$. In real-world applications, the cost of each edge (road) can often be a random variable due to some traffic congestion, and therefore the cost stochastically changes. We assume we can choose an edge (road) each round, and conduct a traffic survey for that edge (road). If we conduct a traffic survey, we can observe a random sample of the cost of the chosen edge. Our goal is to identify the best action, which is a path from the start to the goal nodes.

Although CPE-MAB can be applied to many models which can be formulated as (1), most of the existing works in CPE-MAB (Chen et al. 2014; Wang and Zhu 2022; Gabillon et al. 2016; Chen et al. 2017; Du, Kuroki, and Chen 2021; Chen, Gupta, and Li 2016) assume $\mathcal{A} \subseteq \{0,1\}^d$. This means that the player's objective is to identify the best action which maximizes the sum of the expected rewards. Therefore, although we can apply the existing CPE-MAB methods to the shortest path problem (Sniedovich 2006), top-K arms identification (Kalyanakrishnan and Stone 2010), matching (Gibbons 1985), and spanning trees (Pettie and Ramachandran 2002), we cannot apply them to problems where $\mathcal{A} \subset \mathbb{R}^d$, such as the optimal transport problem (Villani 2008), the knapsack problem (Dantzig and Mazur 2007), and the production planning problem (Pochet and Wolsey 2010). For instance, the optimal transport problem shown in Figure 2 has a real-valued action set $\mathcal{A}$. We have five suppliers and four demanders. Each supplier i has s_i goods to supply. Each demander j wants d_j goods. Each edge μ_{ij} is the cost to transport goods from supplier i to demander j. Our goal is to minimize $\sum_{i=1}^{5} \sum_{j=1}^{4} \pi_{ij} \mu_{ij}$, where $\pi_{ij} (\geq 0)$ is the amount of goods transported to demander j from supplier i. Again, we assume that we can choose an edge (road) each round, and conduct a traffic survey for that edge. Our goal is to identify the best action, which is a transportation plan (matrix) that shows how much

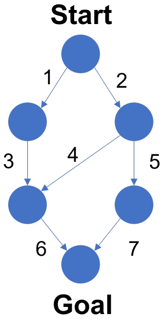

Figure 1: A schematic of the shortest path problem.

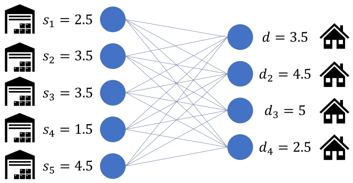

Figure 2: A schematic of the optimal transport problem. One candidate of $\boldsymbol{\pi}$ can be $\boldsymbol{\pi} = \begin{pmatrix} 2.5 & 0 & 0 & 0 \\ 1.0 & 2.5 & 0 & 0 \\ 0 & 2.0 & 1.5 & 0 \\ 0 & 0 & 1.5 & 0 \\ 0 & 0 & 2.0 & 2.5 \end{pmatrix}$.

goods each supplier should send to each demander.

To overcome the limitation of the existing CPE-MAB methods, Nakamura and Sugiyama (2023) has introduced a real-valued CPE-MAB (R-CPE-MAB), where the action set $\mathcal{A} \subset \mathbb{R}^d$. However, it needs an assumption that the size of the action set $\mathcal{A}$ is polynomial in d, which is not satisfied in general since in many combinatorial problems, the action set is exponentially large in d. To cope with this problem, one may leverage algorithms from the *transductive linear bandit* literature (Fiez et al. 2019; Katz-Samuels et al. 2020) for the R-CPE-MAB. In the transductive bandit problem, a player chooses a *probing vector* $\boldsymbol{v}$ from a given finite set $\mathcal{X} \subset \mathbb{R}^d$ each round, and observes $\boldsymbol{\mu}^\top \boldsymbol{v} + \epsilon$, where ϵ is a noise from a certain distribution. Her goal is to identify the best *item* $\boldsymbol{z}^*$ from a finite-sized set $\mathcal{Z} \subset \mathbb{R}^d$, which is defined as $\boldsymbol{z}^* = \arg\max_{\boldsymbol{z} \in \mathcal{Z}} \boldsymbol{\mu}^\top \boldsymbol{z}$. The transductive linear bandit can be seen as a generalization of the R-CPE-MAB since the probing vectors are the standard basis vectors and the items are the actions in the R-CPE-MAB. However, the RAGE algorithm introduced in Fiez et al. (2019) has to enumerate all the items in $\mathcal{Z}$, and therefore, not realistic to apply it when the size of $\mathcal{Z}$ is exponentially large in d. The Peace algorithm (Katz-Samuels et al. 2020) is introduced as an algorithm that can be applied to the CPE-MAB even when the size of $\mathcal{Z}$ is exponentially large in d, but it cannot be applied to the R-CPE-MAB since its subroutine that determines the termination of the algorithm is only valid when $\mathcal{Z} \subset \{0, 1\}^d$.

In this study, we introduce an algorithm named the Generalized Thompson Sampling Explore (GenTS-Explore) algorithm, which can identify the best action in the R-CPE-MAB even when the size of the action set is exponentially large in d. This algorithm can be seen as a generalized version of the Thompson Sampling Explore (TS-Explore) algorithm introduced by Wang and Zhu (2022).

Additionally, we show novel lower bounds of the R-CPE-MAB. One is written explicitly; the other is written implicitly and tighter than the first one. We introduce a hardness measure $\mathbf{H} = \sum_{s=1}^{d} \frac{1}{\Delta_{(s)}^2}$, where $\Delta_{(s)}$ is named *G-Gap*, which can be seen as a generalization of the notion *gap* introduced in the CPE-MAB literature (Chen et al. 2014; Chen, Gupta, and Li 2016; Chen et al. 2017). We show that the sample complexity upper bound of the Gen-TS-Explore algorithm matches the lower bound up to a factor of a problem-dependent constant term.

Problem Formulation

In this section, we formally define the R-CPE-MAB model similar to Chen et al. (2014). Suppose we have d arms, numbered $1, \ldots, d$. Assume that each arm $s \in [d]$ is associated with a reward distribution ϕ_s, where $[d] = \{1, \ldots, d\}$. We assume all reward distributions have an R-sub-Gaussian tail for some known constant $R > 0$. Formally, if X is a random variable drawn from ϕ_s, then, for all $\lambda \in \mathbb{R}$, we have $\mathbb{E}[\exp(\lambda X - \lambda \mathbb{E}[X])] \leq \exp(R^2 \lambda^2 / 2)$. It is known that the family of R-sub-Gaussian tail distributions includes all distributions that are supported on $[0, R]$ and also many unbounded distributions such as Gaussian distributions with variance R^2 (Rivasplata 2012). We denote by $\mathcal{N}(\mu, \sigma^2)$ the Gaussian distribution with mean μ and variance σ^2. Let $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_d)^\top$ denote the vector of expected rewards, where each element $\mu_s = \mathbb{E}_{X \sim \phi_s}[X]$ denotes the expected reward of arm s and $\top$ denotes the transpose. We denote by $T_s(t)$ the number of times arm s is pulled before round t, and by $\hat{\boldsymbol{\mu}}(t) = (\hat{\mu}_1(t), \ldots, \hat{\mu}_d(t))^\top$ the vector of sample means of each arm before round t.

With a given $\boldsymbol{\nu}$, let us consider the following linear optimization problem:

$$\begin{array}{ll} \text{maximize}_{\boldsymbol{\pi}} & \boldsymbol{\nu}^\top \boldsymbol{\pi} \\ \text{subject to} & \boldsymbol{\pi} \in \mathcal{C} \subset \mathbb{R}^d, \end{array} \tag{2}$$

where $\mathcal{C}$ is a problem-dependent feasible region. For any $\boldsymbol{\nu} \in \mathbb{R}^d$, we denote $\boldsymbol{\pi}^{\boldsymbol{\nu}, \mathcal{C}}$ as the optimal solution of (2). Then, we define the action set $\mathcal{A}$ as the set of vectors that contains optimal solutions of (2) for any $\boldsymbol{\nu}$, i.e.,

$$\mathcal{A} = \left\{ \boldsymbol{\pi}^{\boldsymbol{\nu}, \mathcal{C}} \in \mathbb{R}^d \mid \forall \boldsymbol{\nu} \in \mathbb{R}^d \right\}. \tag{3}$$

Note that $|\mathcal{A}|$ could be exponentially large in d. The player's objective is to identify $\boldsymbol{\pi}^* = \arg\max_{\boldsymbol{\pi} \in \mathcal{A}} \boldsymbol{\mu}^\top \boldsymbol{\pi}$ by playing the following game. At the beginning of the game, the action set $\mathcal{A}$ is revealed. Then, the player pulls an arm over a sequence of rounds; in each round t, she pulls an arm $p_t \in [d]$ and observes a reward sampled from the associated reward distribution ϕ_{p_t}. The player can stop the game at any round. She needs to guarantee that $\Pr[\boldsymbol{\pi}_{\text{out}} \neq \boldsymbol{\pi}^*] \leq \delta$ for a given confidence parameter δ. For any $\delta \in (0, 1)$, we call an algorithm $\mathbb{A}$ a δ-correct algorithm if, for any expected reward $\boldsymbol{\mu} \in \mathbb{R}$, the probability of the error of $\mathbb{A}$ is at most δ, i.e., $\Pr[\boldsymbol{\pi}_{\text{out}} \neq \boldsymbol{\pi}^*] \leq \delta$. The learner's performance is evaluated by her *sample complexity*, which is the round she terminated the game. We assume $\boldsymbol{\pi}^*$ is unique.

Technical Assumptions

To cope with the exponential largeness of the action set, we make two mild assumptions for our R-CPE-MAB model. The first one is the existence of the *offline oracle*, which computes $\boldsymbol{\pi}^*(\boldsymbol{\nu}) = \arg\max_{\boldsymbol{\pi} \in \mathcal{A}} \boldsymbol{\nu}^\top \boldsymbol{\pi}$ in polynomial or pseudo-polynomial time once $\boldsymbol{\nu}$ is given. We write $\text{Oracle}(\boldsymbol{\nu}) =$

$\boldsymbol{\pi}^*(\boldsymbol{\nu})$. This assumption is relatively mild since in linear programming, we have the network simplex algorithm (Nelder and Mead 1965) and interior points methods (Karmarkar 1984), whose computational complexities are both polynomials in d. Moreover, if we consider the knapsack problem, though the knapsack problem is NP-complete (Garey and Johnson 1979) and is unlikely that it can be solved in polynomial time, it is well known that we can solve it in pseudo-polynomial time if we use dynamic programming (Kellerer, Pferschy, and Pisinger 2004; Fujimoto 2016). In some cases, it may be sufficient to use this dynamic programming algorithm as the offline oracle in the R-CPE-MAB.

The second assumption is that the set of possible outputs of the offline oracle is finite-sized. This assumption also holds in many combinatorial optimization problems. For instance, no matter what algorithm is used to compute the solution to the knapsack problem, the action set is a finite set of integer vectors, so this assumption holds. Also, in linear programming problems such as the optimal transport problem (Villani 2008) and the production planning problem (Pochet and Wolsey 2010), it is well known that the solution is on a vertex of the feasible region, and therefore, the set of candidates of solutions for optimization problem (1) is finite.

Lower Bound of R-CPE-MAB

In this section, we discuss sample complexity lower bounds of R-CPE-MAB. In Theorem 1, we show a sample complexity lower bound which is derived explicitly. In Theorem 2, we show another lower bound, which is only written in an implicit form but is tighter than that in Theorem 1.

In our analysis, we have several key quantities that are useful to discuss the sample complexity upper bounds. First, we define $\boldsymbol{\pi}^{(s)}$ as follows:

$$\boldsymbol{\pi}^{(s)} = \mathop{\arg\min}_{\boldsymbol{\pi}\in\mathcal{A}\setminus\{\boldsymbol{\pi}^*\}} \frac{\boldsymbol{\mu}^\top(\boldsymbol{\pi}^* - \boldsymbol{\pi})}{|\pi^*_s - \pi_s|}. \tag{4}$$

Intuitively, among the actions whose s-th element is different from $\boldsymbol{\pi}^*$, $\boldsymbol{\pi}^{(s)}$ is the one that is the most difficult to confirm its optimality. We define a notion named *G-gap* which is formally defined as follows:

$$\begin{aligned}\Delta_{(s)} &= \frac{\boldsymbol{\mu}^\top(\boldsymbol{\pi}^* - \boldsymbol{\pi}^{(s)})}{|\pi^*_s - \pi^{(s)}_s|} \\ &= \min_{\boldsymbol{\pi}\in\mathcal{A}\setminus\{\boldsymbol{\pi}^*\}} \frac{\boldsymbol{\mu}^\top(\boldsymbol{\pi}^* - \boldsymbol{\pi})}{|\pi^*_s - \pi_s|}. \end{aligned}\tag{5}$$

G-gap can be seen as a natural generalization of *gap* introduced in the CPE-MAB literature (Chen et al. 2014; Chen, Gupta, and Li 2016; Chen et al. 2017). Then, we denote the sum of inverse squared gaps by

$$\begin{aligned}\mathbf{H} &= \sum_{s=1}^{d}\left(\frac{1}{\Delta_{(s)}}\right)^2 \\ &= \sum_{s=1}^{d} \max_{\boldsymbol{\pi}\in\mathcal{A}\setminus\{\boldsymbol{\pi}^*\}} \frac{|\pi^*_s - \pi_s|^2}{\left((\boldsymbol{\pi}^* - \boldsymbol{\pi})^\top\boldsymbol{\mu}\right)^2},\end{aligned}$$

which we define as a hardness measure of the problem instance in R-CPE-MAB. In Theorem 1, we show that $\mathbf{H}$ appears in a sample complexity lower bound of R-CPE-MAB. Therefore, we expect that this quantity plays an essential role in characterizing the difficulty of the problem instance.

Explicit Form of a Sample Complexity Lower Bound

Here, we show a sample complexity lower bound of the R-CPE-MAB which is written in an explicit form.

Theorem 1. *Fix any action set $\mathcal{A} \subset \mathbb{R}^d$ and any vector $\boldsymbol{\mu} \in \mathbb{R}^d$. Suppose that, for each arm $s \in [d]$, the reward distribution ϕ_s is given by $\phi_s = \mathcal{N}(\mu_s, 1)$. Then, for any $\delta \in \left(0, \frac{e^{-16}}{4}\right)$ and any δ-correct algorithm $\mathbb{A}$, we have*

$$\mathbb{E}\left[T\right] \geq \frac{1}{16}\mathbf{H}\log\left(\frac{1}{4\delta}\right), \tag{6}$$

where T denotes the total number of arm pulls by algorithm $\mathbb{A}$.

Theorem 1 can be seen as a natural generalization of the result in ordinary CPE-MAB shown in Chen et al. (2014). In the CPE-MAB literature, the hardness measure $\mathbf{H}'$ is defined as follows (Chen et al. 2014; Wang and Zhu 2022; Chen et al. 2017):

$$\mathbf{H}' = \sum_{s=1}^{d}\left(\frac{1}{\Delta_s}\right)^2, \tag{7}$$

where

$$\Delta_s = \min_{\boldsymbol{\pi}\in\{\boldsymbol{\pi}\in\mathcal{A}\ |\ \pi_s\neq\pi^*_s\}} \boldsymbol{\mu}^\top(\boldsymbol{\pi}^* - \boldsymbol{\pi}). \tag{8}$$

Below, we discuss why the hardness measure in R-CPE-MAB uses $\Delta_{(s)}$ not Δ_s.

Suppose we have two bandit instances $\mathcal{B}_1$ and $\mathcal{B}_2$. In $\mathcal{B}_1$, $\mathcal{A}_1 = \left\{(100, 0)^\top, (0, 100)^\top\right\}$ and $\boldsymbol{\mu}_1 = (\mu_{1,1}, \mu_{1,2}) = (0.011, 0.01)^\top$. In $\mathcal{B}_2$, $\mathcal{A}_2 = \left\{(1, 0)^\top, (0, 1)^\top\right\}$ and $\boldsymbol{\mu}_2 = (\mu_{2,1}, \mu_{2,2}) = (0.1, 0.11)^\top$. We assume that, for both instances, the arms are equipped with Gaussian distributions with unit variance. Also, for any $i \in \{1, 2\}$ and $s \in \{1, 2\}$, let us denote by $T_{i,s}(t)$ the number of times arm s is pulled in the bandit instance $\mathcal{B}_i$ in round t. Let us consider the situation where $T_{1,1}(t) = T_{2,1}(t)$ and $T_{1,2}(t) = T_{2,2}(t)$, and we have prior knowledge that $\mu_{1,1} \in [\hat{\mu}_{1,1} - \sigma_1, \hat{\mu}_{1,1} + \sigma_1]$, $\mu_{1,2} \in [\hat{\mu}_{1,2} - \sigma_2, \hat{\mu}_{1,2} + \sigma_2]$, $\mu_{2,1} \in [\hat{\mu}_{2,1} - \sigma_1, \hat{\mu}_{2,1} + \sigma_1]$, and $\mu_{2,2} \in [\hat{\mu}_{2,2} - \sigma_2, \hat{\mu}_{2,2} + \sigma_2]$. Here, σ_1 and σ_2 are some confidence bounds on the rewards of arms, which may be derived by some concentration inequality. Note that they depend only on the number of times the arm is pulled, and that the confidence bound for each arm is the same in both instances since $T_{1,1}(t) = T_{2,1}(t)$ and $T_{1,2}(t) = T_{2,2}(t)$.

We can see that $\mathbf{H}'$ are the same in both $\mathcal{B}_1$ and $\mathcal{B}_2$, which implies that the difficulty in identifying the best actions is the same in $\mathcal{B}_1$ and $\mathcal{B}_2$. However, this is not true since when we estimate the reward of actions in $\mathcal{B}_1$, the confidence bound will be amplified by 100, and therefore, we are far less confident to determine the best action in $\mathcal{B}_1$ than $\mathcal{B}_2$. On the other hand, $\mathbf{H}$ reflects this fact. $\mathbf{H}$ in $\mathcal{B}_1$ is 10000 larger than that of $\mathcal{B}_2$, which implies that identifying the best action in $\mathcal{B}_1$ is much more difficult than $\mathcal{B}_2$.

Implicit Form of a Lower Bound

Here, in Theorem 2, we show that we can generalize the tightest lower bound in the CPE-MAB literature shown in Chen et al. (2017) for the R-CPE-MAB.

Theorem 2. *For any* $\delta \in (0, 0.1)$ *and a* δ*-correct algorithm* $\mathbb{A}$*,* $\mathbb{A}$ *will pull arms* $\Omega(\mathrm{Low}(\mathcal{A}) \log \frac{1}{\delta})$ *times, where* $\mathrm{Low}(\mathcal{A})$ *is the optimal value of the following mathematical program:*

$$\begin{aligned} &\text{minimize} && \sum_{s=1}^{d} \tau_s \\ &\text{subject to} && \forall \boldsymbol{\pi} \in \mathcal{A}, \sum_{s \in \boldsymbol{\pi}^* \diamond \boldsymbol{\pi}} \frac{\left|\pi_s^* - \pi_s\right|^2}{\tau_s} \leq \Delta^2_{\boldsymbol{\pi}^*, \boldsymbol{\pi}} \\ & && \tau_s > 0, \forall s \in [d], \end{aligned} \tag{9}$$

where $\boldsymbol{\pi}^* \diamond \boldsymbol{\pi} = \{s \in [d] \mid \pi^*_s \neq \pi_s\}$ *and* $\Delta_{\boldsymbol{\pi}^*, \boldsymbol{\pi}} = \boldsymbol{\mu}^\top (\boldsymbol{\pi}^* - \boldsymbol{\pi})$.

In the appendix, we show that the lower bound in Theorem 2 is no weaker than that in Theorem 1 by showing $\mathrm{Low}(\mathcal{A}) \geq \mathbf{H}$.

This lower bound is exactly equal to the lower bound in Fiez et al. (2019) for the transductive bandit, which is written as follows:

$$\rho_* \log \left(\frac{1}{\delta} \right), \tag{10}$$

where

$$\rho_* = \min_{\boldsymbol{\lambda} \in \Pi_d} \max_{\boldsymbol{\pi} \in \mathcal{A} \setminus \{\boldsymbol{\pi}^*\}} \frac{\sum_{s=1}^{d} \frac{|\pi_s^* - \pi_s|^2}{\lambda_s}}{\Delta^2_{\boldsymbol{\pi}^*, \boldsymbol{\pi}}}. \tag{11}$$

GenTS-Explore Algorithm

In this section, we introduce an algorithm named the Generalized Thompson Sampling Explore (GenTS-Explore) algorithm, which can identify the best action in the R-CPE-MAB even when the size of the action set $\mathcal{A}$ is exponentially large in d. We first explain what the GenTS-Explore algorithm is doing at a high level. Then, we show a sample complexity upper bound of it.

Outline of the GenTS-Explore Algorithm

Here, we show what the GenTS-Explore algorithm is doing at a high level (Algorithm 1).

The GenTS-Explore algorithm can be seen as a modified version of the TS-Explore algorithm introduced in Wang and Zhu (2022). At each round t, it outputs $\hat{\boldsymbol{\pi}}(t) = \mathrm{Oracle}(\hat{\boldsymbol{\mu}}(t))$, which is the *empirically best* action (line 7). Then, for any $s \in [d]$, it draws $M(\delta, q, t) \triangleq \left\lceil \frac{1}{q} (\log 12 |\mathcal{A}|^2 t^2 / \delta) \right\rceil$ random samples $\left\{\theta_s^k\right\}_{k=1}^{M(\delta,q,t)}$ independently from a Gaussian distribution $\mathcal{N}\left(\hat{\mu}_s(t), \frac{C(\delta,q,t)}{T_s(t))}\right)$, and $C(\delta, q, t) \triangleq \frac{4R^2 \log(12|\mathcal{A}|^2 t^2/\delta)}{\phi^2(q)}$. Intuitively, $\{\boldsymbol{\theta}^k(t)\}_{k=1}^{M(\delta,q,t)}$ is a set of possible values that the true reward vector $\boldsymbol{\mu}$ can take. Then, it computes $\tilde{\boldsymbol{\pi}}^k(t) = \mathrm{Oracle}(\boldsymbol{\theta}^k(t))$ for all k, where $\boldsymbol{\theta}^k(t) = \left(\theta_1^k(t), \ldots, \theta_d^k(t)\right)$. We can say that we estimate the true reward gap $\boldsymbol{\mu}^\top \left(\tilde{\boldsymbol{\pi}}^k(t) - \hat{\boldsymbol{\pi}}(t)\right)$ by computing $\boldsymbol{\theta}^k(t)^\top \left(\tilde{\boldsymbol{\pi}}^k(t) - \hat{\boldsymbol{\pi}}(t)\right)$ for each $k \in [M(\delta, q, t)]$. If all the actions $\tilde{\boldsymbol{\pi}}^k(t)$'s are the same as $\hat{\boldsymbol{\pi}}(t)$, we output $\hat{\boldsymbol{\pi}}(t)$ as the best action. Otherwise, we focus on $\tilde{\boldsymbol{\pi}}^{k_t^*}(t)$, where $k_t^* = \arg\max_{k \in [M(\delta,q,t)]} \boldsymbol{\theta}^k(t)^\top \left(\tilde{\boldsymbol{\pi}}^k(t) - \hat{\boldsymbol{\pi}}(t)\right)$. We can say that $\tilde{\boldsymbol{\pi}}^{k_t^*}(t)$ is potentially the best action.

Algorithm 1: GenTS-Explore Algorithm

```
1: Input: Confidence level $\delta$, $q \in [\delta, 0.1]$, $t \leftarrow 0$, $\boldsymbol{T}(0) = (T_1(0), \ldots, T_d(0))^\top = (0, \ldots, 0)^\top$
2: Output: Action $\boldsymbol{\pi}_{\mathrm{out}} \in \mathcal{A}$
3: // Initialization
4: Pull each arm once, and update their number of pulls $T_i$'s and the $\hat{\mu}_s(t)$
5: $t \leftarrow d$
6: while true do
7:    $\hat{\boldsymbol{\pi}}(t) \leftarrow \mathrm{Oracle}(\hat{\boldsymbol{\mu}}(t))$
8:    for $k = 1, \ldots, M(\delta, q, t)$ do
9:       For each arm $s$, draw $\theta_s^k(t)$ independently from distribution $\mathcal{N}\left(\hat{\mu}_s(t), \frac{C(\delta,q,t)}{T_s(t)}\right)$
10:      $\boldsymbol{\theta}^k(t) \leftarrow \left(\theta_1^k(t), \ldots, \theta_d^k(t)\right)$
11:      $\tilde{\boldsymbol{\pi}}^k(t) \leftarrow \mathrm{Oracle}(\boldsymbol{\theta}^k(t))$
12:      $\tilde{\Delta}_t^k \leftarrow \boldsymbol{\theta}^k(t)^\top \left(\tilde{\boldsymbol{\pi}}^k(t) - \hat{\boldsymbol{\pi}}(t)\right)$
13:   end for
14:   if $\forall 1 \leq k \leq M(\delta, q, t), \tilde{\boldsymbol{\pi}}^k(t) = \hat{\boldsymbol{\pi}}(t)$ then
15:      Return: $\hat{\boldsymbol{\pi}}(t)$
16:   else
17:      $k_t^* \leftarrow \arg\max_k \tilde{\Delta}_t^k$, $\tilde{\boldsymbol{\pi}}(t) \leftarrow \tilde{\boldsymbol{\pi}}^{k_t^*}(t)$
18:      Pull arm $p_t$ according to (12) or (15),and update $T_{p_t}$ and $\hat{\mu}_{p_t}(t)$
19:      $t \leftarrow t + 1$
20:   end if
21: end while
```

Then, the most essential question is: "Which arm should we pull in round t, once we obtain the empirically best action $\hat{\boldsymbol{\pi}}(t)$ and a potentially best action $\tilde{\boldsymbol{\pi}}^{k^*}(t)$?" We discuss this below.

Arm Selection Strategies Here, we discuss which arm to pull at round t, once we obtain the empirically best action $\hat{\boldsymbol{\pi}}(t)$ and a potentially best action $\tilde{\boldsymbol{\pi}}^{k_t^*}(t)$. For the ordinary CPE-MAB, the arm selection strategy in Wang and Zhu (2022) was to pull the following arm:

$$p_t^{\mathrm{naive}} = \mathop{\arg\min}_{s \in \left\{s \in [d] \;\mid\; \hat{\pi}_s(t) \neq \tilde{\pi}_s^{k^*}(t)\right\}} T_s(t). \tag{12}$$

Therefore, one candidate of an arm selection strategy is to naively pull the arm defined in (12). We call this the *naive arm selection strategy*.

Next, we consider another arm selection strategy as follows. We want to pull the arm that is most "informative" to discriminate whether $\tilde{\boldsymbol{\pi}}(t)$ is a better action than $\tilde{\boldsymbol{\pi}}^{k_t^*}(t)$ or not. In other words, we want to pull the arm that is most "informative" to estimate the true gap $\boldsymbol{\mu}^\top \left(\tilde{\boldsymbol{\pi}}^{k_t^*}(t) - \hat{\boldsymbol{\pi}}(t)\right)$. If it is less than 0, $\hat{\boldsymbol{\pi}}(t)$ is better, and if it is greater than 0, $\tilde{\boldsymbol{\pi}}^{k_t^*}$ is better. To discuss this more quantitatively, let us assume that $\boldsymbol{\theta}^{k_t^*}(t) \approx \hat{\boldsymbol{\mu}}(t)$. From Hoeffding's inequality (Luo 2017), we obtain the following:

$$\begin{aligned} & \Pr\left[\left|\left(\boldsymbol{\mu} - \boldsymbol{\theta}^{k_t^*}(t)\right)^\top \left(\tilde{\boldsymbol{\pi}}^{k_t^*}(t) - \hat{\boldsymbol{\pi}}(t)\right)\right| \geq \epsilon\right] \\ \approx \ & \Pr\left[\left|\left(\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}(t)\right)^\top \left(\tilde{\boldsymbol{\pi}}^{k_t^*}(t) - \hat{\boldsymbol{\pi}}(t)\right)\right| \geq \epsilon\right] \\ \leq \ & 2\exp\left(-\frac{\epsilon^2}{2\sum_{s=1}^{d} \frac{\left|\tilde{\pi}_s^{k_t^*}(t) - \hat{\pi}_s(t)\right|^2}{T_s(t)} R^2}\right), \end{aligned} \tag{13}$$

where $\epsilon > 0$. (13) shows that if we make $\sum_{s=1}^{d} \frac{\left|\tilde{\pi}_s^{k_t^*}(t) - \hat{\pi}_s(t)\right|^2}{T_s(t)}$ small, $\tilde{\Delta}_t^{k_t^*} = \boldsymbol{\theta}^{k_t^*}(t)^\top \left(\tilde{\boldsymbol{\pi}}^{k_t^*}(t) - \hat{\boldsymbol{\pi}}(t)\right)$ will become close to the true gap $\boldsymbol{\mu}^\top \left(\tilde{\boldsymbol{\pi}}^{k_t^*}(t) - \hat{\boldsymbol{\pi}}(t)\right)$.

Since we want to estimate the true gap accurately as soon as possible, we pull arm p_t^{R} that makes $\sum_{s=1}^{d} \frac{\left|\tilde{\pi}_s^{k_t^*}(t) - \hat{\pi}_s(t)\right|^2}{T_s(t)}$ the smallest, which is defined as follows:

$$p_t^{\mathrm{R}} = \operatorname*{arg\,min}_{e \in [d]} \sum_{s=1}^{d} \frac{\left|\tilde{\pi}_s^{k_t^*}(t) - \hat{\pi}_s(t)\right|^2}{T_s(t) + \mathbf{1}[s = e]}, \tag{14}$$

where $\mathbf{1}[\cdot]$ denotes the indicator function. Then, the following proposition holds.

Proposition 3. p_t^{R} *in (14) can be written as follows:*

$$p_t^{\mathrm{R}} = \operatorname*{arg\,max}_{s \in [d]} \frac{\left|\tilde{\pi}_s^{k_t^*}(t) - \hat{\pi}_s(t)\right|^2}{T_s(t)(T_s(t) + 1)}. \tag{15}$$

We show the proof in the appendix. We call pulling the arm defined in (15) the *R-CPE-MAB arm selection strategy*. (15) implies that the larger $\left|\tilde{\pi}_s^{k_t^*}(t) - \hat{\pi}_s(t)\right|$ is, the more we need to pull arm s. Similar to the discussion in the previous section, this is because if $\left|\tilde{\pi}_s^{k_t^*}(t) - \hat{\pi}_s(t)\right|$ is large, the uncertainty of arm s is amplified largely when we compute $\tilde{\Delta}_t^{k_t^*} = \boldsymbol{\theta}^{k_t^*}(t)^\top \left(\tilde{\boldsymbol{\pi}}^{k_t^*}(t) - \hat{\boldsymbol{\pi}}(t)\right)$. Therefore, we have to pull arm s many times to make the $\frac{C(\delta, q, t)}{T_s(t)}$ small, which is the variance of θ_s^k, to gain more confidence about the reward of arm s.

Also, the *R-CPE-MAB arm selection strategy* is equivalent to the *naive arm selection strategy* in CPE-MAB, since when $\mathcal{A} \subset \{0, 1\}^d$,

$$\begin{aligned} p_t^{\mathrm{R}} &= \operatorname*{arg\,max}_{s \in [d]} \frac{\left|\tilde{\pi}_s^{k_t^*}(t) - \hat{\pi}_s(t)\right|^2}{T_s(t)(T_s(t) + 1)} \\ &= \operatorname*{arg\,max}_{s \in \left\{s \in [d] \ | \ \tilde{\pi}_s^{k_t^*}(t) \neq \hat{\pi}_s(t)\right\}} \frac{1}{T_s(t)(T_s(t) + 1)} \\ &= \operatorname*{arg\,min}_{s \in \left\{s \in [d] \ | \ \tilde{\pi}_s^{k_t^*}(t) \neq \hat{\pi}_s(t)\right\}} T_s(t). \\ &= p_t^{\mathrm{naive}}. \end{aligned} \tag{16}$$

Therefore, we can say that the *R-CPE-MAB arm selection strategy* is a generalization of the arm selection strategy in Wang and Zhu (2022).

Sample Complexity Upper Bounds of the GenTS-Explore Algorithm

Here, we show sample complexity upper bounds of the GenTS-Explore algorithm when we use the two arm selection strategies: the *naive arm selection strategy* shown in (12) and the *R-CPE-MAB arm selection strategy* shown in (15), respectively.

First, in Theorem 4, we show a sample complexity upper bound of the *naive arm selection strategy*.

Theorem 4. *For $q \in [\delta, 0.1]$, with probability at least $1 - \delta$, the GenTS-Explore algorithm with the naive arm sampling strategy will output the best action $\boldsymbol{\pi}^*$ with sample complexity upper bounded by*

$$\mathcal{O}\left(R^2 \mathbf{H}^{\mathrm{N}} \frac{\left(\log\left(|\mathcal{A}|\, \mathbf{H}^{\mathrm{N}}\right) + \log\frac{1}{\delta}\right)^2}{\log\frac{1}{q}}\right), \tag{17}$$

where $\mathbf{H}^{\mathrm{N}} = \sum_{s=1}^{d} \frac{U_s}{\Delta_{(s)}^2}$ *and* $U_s = \max_{\boldsymbol{\pi}' \in \mathcal{A}, \boldsymbol{\pi} \in \{\boldsymbol{\pi} \in \mathcal{A} \ | \ \pi_s^* \neq \pi_s\}} \frac{1}{|\pi_s^* - \pi_s|^2} \sum_{e=1}^{d} |\pi_e - \pi_e'|^2$.

Specifically, if we choose $q = \delta$, then the complexity upper bound is

$$\mathcal{O}\left(R^2 \mathbf{H}^{\mathrm{N}} \left(\log\frac{1}{\delta} + \log^2\left(|\mathcal{A}|\, \mathbf{H}^{\mathrm{N}}\right)\right)\right). \tag{18}$$

Next, in Theorem 5, we show a sample complexity upper bound of the *R-CPE-MAB arm selection strategy*.

Theorem 5. *For $q \in [\delta, 0.1]$, with probability at least $1 - \delta$, the GenTS-Explore algorithm with the R-CPE-MAB arm sampling strategy will output the best action $\boldsymbol{\pi}^*$ with sample complexity upper bounded by*

$$\mathcal{O}\left(R^2 \mathbf{H}^{\mathrm{R}} \frac{\left(\log\left(|\mathcal{A}|\, \mathbf{H}^{\mathrm{R}}\right) + \log\frac{1}{\delta}\right)^2}{\log\frac{1}{q}}\right), \tag{19}$$

where $\mathbf{H}^{\mathrm{R}} = \sum_{s=1}^{d} \frac{V_s}{\Delta_{(s)}^2}$ *and* $V_s = \max_{\boldsymbol{\pi}' \in \mathcal{A}, \boldsymbol{\pi} \in \{\boldsymbol{\pi} \in \mathcal{A} \ | \ \pi_s^* \neq \pi_s\}} \frac{\left|\pi_s - \pi_s'\right|}{|\pi_s^* - \pi_s|^2} \sum_{e=1}^{d} |\pi_e - \pi_e'|$.

Specifically, if we choose $q = \delta$, then the complexity upper bound is

$$\mathcal{O}\left(R^2\mathbf{H}^{\mathrm{R}}\left(\log\frac{1}{\delta}+\log^2\left(|\mathcal{A}|\,\mathbf{H}^{\mathrm{R}}\right)\right)\right). \tag{20}$$

Comparison to the Lower Bounds Let us define $U = \max_{s\in[d]} U_s$ and $V = \max_{s\in[d]} V_s$. Then, the sample complexity upper bound of the naive arm selection strategy is $\mathcal{O}\left(U\mathbf{H}\log\left(\frac{1}{\delta}\right)\right)$ and that of the R-CPE-MAB arm selection strategy is $\mathcal{O}\left(V\mathbf{H}\log\left(\frac{1}{\delta}\right)\right)$. Therefore, regardless of which arm selection strategy is used, the sample complexity upper bound of the GenTS-Explore algorithm matches the lower bound shown in (6) up to a problem-dependent constant factor.

Comparison between the Naive and R-CPE-MAB Arm Selection Strategies In general, whether the *R-CPE-MAB arm selection strategy* has a tighter upper bound than the *naive arm selection strategy* or not depends on the problem instance. Let us consider one situation in which the R-CPE-MAB arm selection strategy may be a better choice than the naive arm selection strategy. Suppose $\mathcal{A} = \{(100,0,0)^\top, (0,1,1)^\top\}$ and $\boldsymbol{\pi}^* = (100,0,0)^\top$. Then, $U_1 = 1.0002$, $U_2 = 10002$, and $U_3 = 10002$. On the other hand, $V_1 = 1.02$, $V_2 = 102$, and $V_3 = 102$. We can see that U_2 and U_3 are extremely larger than V_2 and V_3, respectively, and therefore $\mathbf{H}^{\mathrm{R}}$ is much smaller than $\mathbf{H}^{\mathrm{N}}$. Eventually, the sample complexity upper bound of the naive arm selection strategy will be looser than that of the R-CPE-MAB arm selection strategy.

Comparison with Existing Works in the Ordinary CPE-MAB In the ordinary CPE-MAB, where $\mathcal{A} \subseteq \{0,1\}^d$, a key notion called *width* appears in the upper bound of some existing algorithms (Chen et al. 2014; Wang and Zhu 2022), which is defined as follows:

$$\mathrm{width} = \max_{\boldsymbol{\pi},\boldsymbol{\pi}'\in\mathcal{A}}\sum_{s=1}^{d}\left|\pi_s - \pi'_s\right|. \tag{21}$$

The following proposition implies that both U and V can be seen as generalizations of the notion *width*.

Proposition 6. *Let $U = \max_{s\in[d]} U_s$ and $V = \max_{s\in[d]} V_s$. In the ordinary CPE-MAB, where $\mathcal{A} \subseteq \{0,1\}^d$, we have*

$$U = V = \mathrm{width}. \tag{22}$$

Next, recall that the GenTS-Explore algorithm is equivalent to the TS-Explore algorithm in the ordinary CPE-MAB, regardless of which arm selection strategy is used. Proposition 7 shows that our upper bound (18) and (20) are both tighter than that shown in Wang and Zhu (2022), which is $\mathcal{O}\left(\mathrm{width}\sum_{s=1}^{d}\frac{1}{\Delta_s^2}\right)$.

Proposition 7. *In the ordinary CPE-MAB, where $\mathcal{A} \subseteq \{0,1\}^d$, we have*

$$\mathbf{H}^{\mathrm{N}} = \sum_{s=1}^{d}\frac{U_s}{\Delta_s^2} \leq \mathrm{width}\sum_{s=1}^{d}\frac{1}{\Delta_s^2}, \tag{23}$$

and

$$\mathbf{H}^{\mathrm{R}} = \sum_{s=1}^{d}\frac{V_s}{\Delta_s^2} \leq \mathrm{width}\sum_{s=1}^{d}\frac{1}{\Delta_s^2}. \tag{24}$$

Experiment

In this section, we conduct experiments on the knapsack (Dantzig and Mazur 2007) and production planning (Pochet and Wolsey 2010) problems and experimentally compare two main arm selection strategies: the naive arm selection strategy and the R-CPE-MAB arm selection strategy. Also, for the knapsack problem, we compare the GenTS-Explore algorithm with other algorithms that can be applied to the R-CPE-MAB.

The Knapsack Problem

Here, we consider the knapsack problem (Dantzig and Mazur 2007), where the action set $\mathcal{A}$ is exponentially large in d in general.

In the knapsack problem, we have d items. Each item $s \in [d]$ has a weight w_s and value v_s. Also, there is a knapsack whose capacity is W in which we put items. Our goal is to maximize the total value of the knapsack not letting the total weight of the items exceed the capacity of the knapsack. Formally, the optimization problem is given as follows:

$$\begin{aligned}&\mathrm{maximize}_{\boldsymbol{\pi}\in\mathcal{A}} && \textstyle\sum_{s=1}^{d} v_s\pi_s\\ &\text{subject to} && \textstyle\sum_{s=1}^{d}\pi_s w_s \leq W,\end{aligned}$$

where π_s denotes the number of item s in the knapsack. Here, the weight of each item is known, but the value is unknown, and therefore has to be estimated. In each time step, the player chooses an item s and gets an observation of value r_s, which can be regarded as a random variable from an unknown distribution with mean v_s.

For our experiment, we generate the weight of each item uniformly from $\{1,2,\ldots,50\}$. For each item s, we generate v_s as $v_s = w_s \times (1+x)$, where x is a sample from $\mathcal{N}(0, 0.1^2)$. We set the capacity of the knapsack at $W = 50$. Each time we choose an item s, we observe a value $v_s + x$ where x is a noise from $\mathcal{N}(0, 0.1^2)$. We set $R = 0.1$. We show the result in Figure 3.

We can say that the R-CPE-MAB arm selection strategy performs better than the naive arm selection strategy since the former needs fewer rounds until termination. In some cases, the sample complexity of the R-CPE-MAB arm selection strategy is only 1/3 to 1/2 that of the naive arm selection strategy.

Next, we compare the GenTS-Explore algorithm with the CombGapE (Nakamura and Sugiyama 2023), RAGE (Randomized Adaptive Gap Elimination) Fiez et al. (2019) and Peace(Katz-Samuels et al. 2020) algorithms. Note that these algorithms have to enumerate all the feasible actions, which is nearly impossible in practice when the size of the action set is exponentially large in d.

We conducted an experiment on a knapsack problem with $d = 5$. We show the results in Table 1. We can see that while the GenTS-Explore algorithm can not outperform the CombGapE algorithm, it outperforms the other two methods.

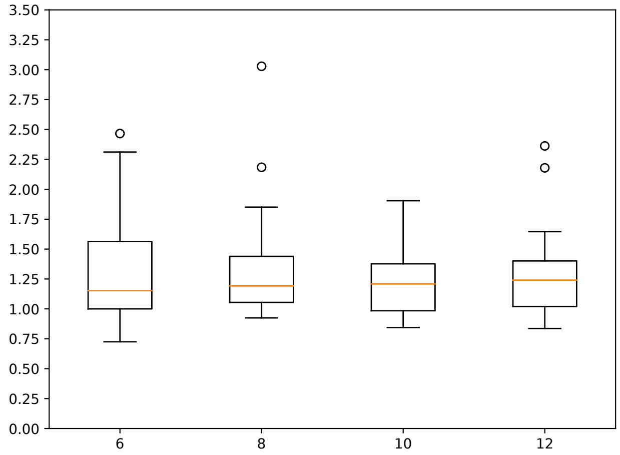

Figure 3: Comparison of the naive arm selection strategy and the R-CPE-MAB arm selection strategy. The vertical axis indicates the number of rounds the former strategy takes to find the best action normalized by the number of rounds the latter strategy takes to find the best action. The horizontal axis indicates the number of items d. We ran experiments 30 times for each setting.

GenTS-Explore	44 ± 40
RAGE (Fiez et al. 2019)	$1.9 \times 1.4^4 \pm 2.7 \times 10^4$
Peace (Katz-Samuels et al. 2020)	66 ± 54

Table 1: The mean and standard deviation of sample complexity normalized by the sample complexity of the CompGapE algorithm.

The Production Planning Problem

Here, we consider the production planning problem (Pochet and Wolsey 2010). In the production planning problem, there are m materials, and these materials can be mixed to make one of d different products. We have a matrix $\boldsymbol{M} \in \mathbb{R}^{m \times d}$, where M_s represents how much material $i \in [m]$ is needed to make product $s \in [d]$. Also, we are given vectors $\boldsymbol{v}^{\max} \in \mathbb{R}^m$ and $\boldsymbol{\mu} \in \mathbb{R}^d$. Then, formally, the optimization problem is given as follows:

$$\begin{aligned} &\text{maximize}_{\boldsymbol{\pi} \in \mathcal{A}} && \boldsymbol{\mu}^\top \boldsymbol{\pi} \\ &\text{subject to} && \boldsymbol{M}\boldsymbol{\pi} \leq \boldsymbol{v}^{\max}, \end{aligned}$$

where the inequality is an element-wise comparison. Intuitively, we want to obtain the optimal vector $\boldsymbol{\pi}^*$ that maximizes the total profit without using more material i than $v_i^{\max}$ for each $i \in [m]$, where π_s^* represents how much product s is produced.

Here, we assume that $\boldsymbol{M}$ and $\boldsymbol{v}^{\max}$ are known, but $\boldsymbol{\mu}$ is unknown, and therefore has to be estimated. In each time step, the player chooses a product s and gets an observation of value r_s, which can be regarded as a random variable from an unknown distribution with mean μ_s.

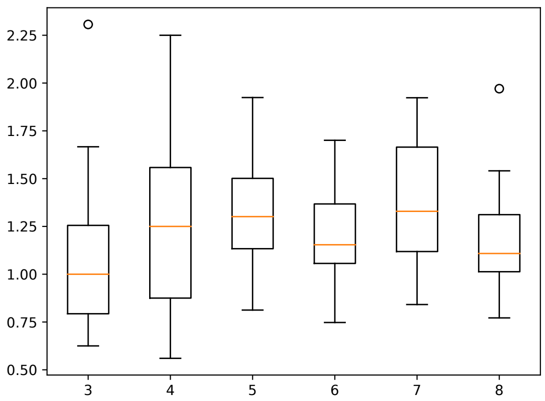

Figure 4: The numbers show the mean and standard deviation of the number of rounds the naive arm selection strategy takes to find the best action normalized by the number of rounds the R-CPE-MAB arm selection strategy takes to find the best action over 15 runs.

For our experiment, we have three materials, i.e., $m = 3$. We set $\boldsymbol{v}^{\max} = (30, 30, 30)^\top$. Also, we generate every element in M uniformly from $\{1, 2, 3, 4\}$. For each product s, we generate μ_s as $\mu_s = \sum_{i=1}^m M_{is} + x$, where x is a random sample from $\mathcal{N}(0, 1)$. Each time we choose a product s, we observe a value $\mu_s + x$ where x is a noise from $\mathcal{N}\left(0, 0.1^2\right)$. We set $R = 0.1$. We show the result in Figure 4. Again, we can see that the R-CPE-MAB arm selection strategy performs better than the naive arm selection strategy since the former needs fewer rounds until termination.

Conclusion

In this study, we studied the R-CPE-MAB. We showed novel lower bounds for R-CPE-MAB by generalizing key quantities in the ordinary CPE-MAB literature. Then, we introduced an algorithm named the GenTS-Explore algorithm, which can identify the best action in R-CPE-MAB even when the size of the action set is exponentially large in d. We showed a sample complexity upper bound of it, and showed that it matches the sample complexity lower bound up to a problem-dependent constant factor. Finally, we experimentally showed that the GenTS-Explore algorithm can identify the best action even if the action set is exponentially large in d.

Acknowledgements

SN was supported by JST SPRING, Grant Number JPMJSP2108. MS was supported by the Institute for AI and Beyond, UTokyo.

References

Audibert, J.-Y.; Bubeck, S.; and Munos, R. 2010. Best Arm Identification in Multi-Armed Bandits. In *The 23rd Conference on Learning Theory*, 41–53.

Bubeck, S.; Munos, R.; and Stoltz, G. 2009. Pure Exploration in Multi-armed Bandits Problems. In *International Conference on Algorithmic Learning Theory*.

Chen, L.; Gupta, A.; and Li, J. 2016. Pure Exploration of Multi-armed Bandit Under Matroid Constraints. In *Proceedings of the 29th Conference on Learning Theory, COLT 2016, New York, USA, June 23-26, 2016*, volume 49 of *JMLR Workshop and Conference Proceedings*, 647–669. JMLR.org.

Chen, L.; Gupta, A.; Li, J.; Qiao, M.; and Wang, R. 2017. Nearly Optimal Sampling Algorithms for Combinatorial Pure Exploration. In *Proceedings of the 2017 Conference on Learning Theory*, volume 65 of *Proceedings of Machine Learning Research*, 482–534. PMLR.

Chen, S.; Lin, T.; King, I.; Lyu, M. R.; and Chen, W. 2014. Combinatorial Pure Exploration of Multi-Armed Bandits. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*, 379–387. Cambridge, MA, USA: MIT Press.

Dantzig, T.; and Mazur, J. 2007. *Number: The Language of Science*. A Plume book. Penguin Publishing Group.

Du, Y.; Kuroki, Y.; and Chen, W. 2021. Combinatorial Pure Exploration with Bottleneck Reward Function. In *Advances in Neural Information Processing Systems*, volume 34, 23956–23967. Curran Associates, Inc.

Fiez, T.; Jain, L.; Jamieson, K.; and Ratliff, L. 2019. *Sequential Experimental Design for Transductive Linear Bandits*. Red Hook, NY, USA: Curran Associates Inc.

Fujimoto, N. 2016. A Pseudo-Polynomial Time Algorithm for Solving the Knapsack Problem in Polynomial Space. In Chan, T.-H. H.; Li, M.; and Wang, L., eds., *Combinatorial Optimization and Applications*, 624–638. Cham: Springer International Publishing. ISBN 978-3-319-48749-6.

Gabillon, V.; Lazaric, A.; Ghavamzadeh, M.; Ortner, R.; and Bartlett, P. 2016. Improved Learning Complexity in Combinatorial Pure Exploration Bandits. In *Proceedings of the 19th International Conference on Artificial Intelligence and Statistics*, volume 51 of *Proceedings of Machine Learning Research*, 1004–1012. Cadiz, Spain: PMLR.

Garey, M. R.; and Johnson, D. S. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences)*. W. H. Freeman, first edition edition. ISBN 0716710455.

Gibbons, A. 1985. *Algorithmic Graph Theory*. Cambridge University Press. ISBN 9780521288811.

Kalyanakrishnan, S.; and Stone, P. 2010. Efficient Selection of Multiple Bandit Arms: Theory and Practice. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*, ICML'10, 511–518. Madison, WI, USA: Omnipress. ISBN 9781605589077.

Karmarkar, N. 1984. A New Polynomial-Time Algorithm for Linear Programming. In *Proceedings of the Sixteenth Annual ACM Symposium on Theory of Computing*, STOC '84, 302–311. New York, NY, USA: Association for Computing Machinery. ISBN 0897911334.

Katz-Samuels, J.; Jain, L.; Karnin, Z.; and Jamieson, K. 2020. An Empirical Process Approach to the Union Bound: Practical Algorithms for Combinatorial and Linear Bandits. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20. Red Hook, NY, USA: Curran Associates Inc. ISBN 9781713829546.

Kellerer, H.; Pferschy, U.; and Pisinger, D. 2004. *Knapsack Problems*. Springer, Berlin, Germany.

Luo, S. 2017. Sub-Gaussian random variable and its properties.

Nakamura, S.; and Sugiyama, M. 2023. An Optimal Algorithm for the Real-Valued Combinatorial Pure Exploration of Multi-Armed Bandit. arXiv:2306.09202.

Nelder, J. A.; and Mead, R. 1965. A simplex method for function minimization. *Computer Journal*, 7: 308–313.

Pettie, S.; and Ramachandran, V. 2002. An Optimal Minimum Spanning Tree Algorithm. *J. ACM*, 49(1): 16–34.

Pochet, Y.; and Wolsey, L. A. 2010. *Production Planning by Mixed Integer Programming*. Springer Publishing Company, Incorporated, 1st edition. ISBN 144192132X.

Rivasplata, O. 2012. Subgaussian random variables : An expository note.

Sniedovich, M. 2006. Dijkstra's algorithm revisited: the dynamic programming connexion. *Control and Cybernetics*, 35: 599–620.

Villani, C. 2008. *Optimal Transport: Old and New*. Grundlehren der mathematischen Wissenschaften. Springer Berlin Heidelberg. ISBN 9783540710509.

Wang, S.; and Zhu, J. 2022. Thompson Sampling for (Combinatorial) Pure Exploration. In *Proceedings of the 39 th International Conference on Machine Learning*. Baltimore, Maryland, USA.