Combinatorial Pure Exploration with Partial or Full-Bandit Linear Feedback

Wei Chen∗
Microsoft Research
weic@microsoft.com

Yihan Du†
IIIS, Tsinghua University
duyh18@mails.tsinghua.edu.cn

Yuko Kuroki
The University of Tokyo, RIKEN
ykuroki@ms.k.u-tokyo.ac.jp

Abstract

In this paper, we propose the novel model of combinatorial pure exploration with partial linear feedback (CPE-PL). In CPE-PL, given a combinatorial action space \( X \subseteq \{0, 1\}^d \), in each round a learner chooses one action \( x \in X \) to play, obtains a random (possibly nonlinear) reward related to \( x \) and an unknown latent vector \( \theta \in \mathbb{R}^d \), and observes a partial linear feedback \( M_x(\theta + \eta) \), where \( \eta \) is a zero-mean noise vector and \( M_x \) is a transformation matrix for \( x \). The objective is to identify the optimal action with the maximum expected reward using as few rounds as possible. We also study the important subproblem of CPE-PL, i.e., combinatorial pure exploration with full-bandit feedback (CPE-BL), in which the learner observes full-bandit feedback (i.e. \( M_x = x^\top \)) and gains linear expected reward \( x^\top \theta \) after each play. In this paper, we first propose a polynomial-time algorithmic framework for the general CPE-PL problem with novel sample complexity analysis. Then, we propose an adaptive algorithm dedicated to the subproblem CPE-BL with better sample complexity. Our work provides a novel polynomial-time solution to simultaneously address limited feedback, general reward function and combinatorial action space including matroids, matchings, and s-t paths.

1 Introduction

The problem of best arm identification (BAI) is the pure-exploration framework in the stochastic multi-armed bandits. In BAI, a learner chooses an arm and observes a reward sampled from an unknown distribution at each step, and then she must return the best arm with the highest expected reward at the end of the exploration phase. This problem abstracts a decision making model with a wide range of applications in the face of uncertainty, and has received much attentions in the literature [2, 6, 13, 16, 17, 24, 27, 29].

In many application domains, it is natural that possible actions have a certain combinatorial structure. For example, each action may be a size-\( k \) subset of keywords in online advertisements [41], or an assignment between workers and tasks in crowdsourcing [37], or a spanning tree in communication networks [23]. To deal with such a combinatorial action space, the model of combinatorial pure exploration of multi-armed bandits (CPE-MB) was first proposed by Chen et al. [14]. In this model, there are \( d \) base arms, and an action that generates reward is a super arm that consists of a subset of base arms. A learner plays a base arm at each step and observes its random feedback, and the goal is to identify the best super arm with the highest sum of the rewards at the end of exploration. They investigated two algorithms called, CLUCB and CSAR for the fixed confidence setting and fixed budget.

∗The authors are listed in alphabetical order.
†This work was mostly done while Yihan Du was an intern at Microsoft Research Asia.
setting, which can work for general combinatorial structures such as matorids, matchings and paths. CPE-MB generalizes the classical BAI and Top-\(k\) identification \([3, 5, 7, 19, 20, 25, 26, 30, 40, 45]\). Recently, a line of work has established near-optimal sampling methods for CPE-MB \([4, 11, 12, 21]\).

However, CPE-MB still has critical limitations in real world scenarios. CPE-MB assumes that the learner can directly play each base arm and observe its outcome, but this might not be allowed due to system constraints or privacy issues. Despite the real-world requirements, there are only a few studies that can avoid such an assumption. Kuroki et al. \([34]\) studied the combinatorial pure exploration with full-bandit linear feedback (CPE-BL), in which the learner pulls a super arm (rather than base arm) and only observes a total sum of random rewards from the pulled super arm. They designed a polynomial-time approximation algorithm for a 0-1 quadratic programming and proposed the first static allocation algorithm for CPE-BL, wherein arm selection ratio is predetermined and independent of any past observation. Rejwan and Mansour \([39]\) also investigated CPE-BL for the Top-\(k\) case with the aid of a Hadamard matrix and designed an adaptive algorithm based on Successive Accept Reject strategy.

In this paper, we further study the more challenging problem, combinatorial pure exploration with partial-monitoring linear feedback (CPE-PL), which simultaneously models limited feedback, general (possibly nonlinear) reward and combinatorial action space. In CPE-PL, given a combinatorial action space \(X \subseteq \{0, 1\}^d\), where each dimension corresponds to a base arm and each action \(x \in X\) can also be viewed as a super arm that contains those dimensions with coordinate 1. In each round the learner chooses an action (super arm) \(x_t \in X\) to play and observes a random partial linear feedback with expectation of \(M_{x_t}\theta\), where \(M_{x_t}\) is a transformation matrix for \(x_t\) and \(\theta \in \mathbb{R}^d\) is an unknown environment vector. The learner also gains a random (possibly nonlinear) reward related to \(x_t\) and \(\theta\), which may not be part of the feedback and thus may not be directly observed. Given a confidence level \(\delta\), the objective is to identify the optimal action with the maximum expected reward with probability at least \(1 - \delta\), using as few rounds as possible. CPE-PL framework includes CPE-BL as its important subproblem: in CPE-BL, the learner observes full-bandit feedback (i.e. \(M_{x} = x^\top\)) and gains linear reward (with expectation of \(x^\top \theta\)) after each play. The model of CPE-PL appears in many practical scenarios, including:

- **Learning to rank.** Suppose that a company (learner) wishes to recommend their products to users by presenting the ranked list of items. Collecting a large amount of data on the relevance of all items might be infeasible, but the relevance of only a small subset of items which are highly-ranked (or top-ranked item) is reasonable to obtain \([8, 9, 10]\). In this scenario, the learner selects a ranked list of entire items at each step, and observes random partial linear feedback on the relevance of highly-ranked \(d' \ll d\) items. The objective is to identify the best ranking of their whole items as soon as possible.

- **Task assignments in crowdsourcing.** Suppose that an employer wishes to assign crowd-workers to tasks with high quality performance. It might be costly for the employer and workers to provide task performance feedback for all tasks \([37]\), and also privacy issues may arise. In this scenario, the learner sequentially chooses an assignment from workers to tasks and observes random partial linear feedback on a small subset of completed tasks. The objective is to find the matching between workers and tasks with the highest performance as soon as possible.

Note that CPE-BL, the subproblem of CPE-PL, can be regarded as an instance of the best arm identification in linear bandits (BAI-LB), which has received increasing attention \([2, 18, 28, 35, 42, 43, 44]\). In linear bandits, each arm has its own feature \(x \in \mathbb{R}^d\), while in CPE-BL, each super arm is a 0-1 vector \(x \in \{0, 1\}^d\). Soare et al. \([42]\) addressed the BAI-LB in the fixed confidence setting and first provided the static allocation algorithm for BAI-LB by introducing the interesting connection between BAI-LB and G-optimal experimental design \([38]\). Tao et al. \([43]\) analyzed the novel randomized estimator based on the convex relaxation of G-optimal design, and devised the adaptive algorithm whose sample complexity depends linearly on the dimension \(d\). Xu et al. \([44]\) proposed a fully adaptive algorithm inspired by UGapE \([19]\). Karnin \([28]\) analyzed the explore-verify
algorithms for several settings of BAI including linear, dueling, unimodal, and graphical bandits. Fiez et al. [18] introduced the transductive BAI-LB, and proposed the first non-asymptotic algorithm that nearly achieves the information-theoretic lower bound. Unfortunately, any existing algorithms for BAI-LB cannot solve CPE-PL in polynomial time because its time complexity has polynomial dependency on the number of actions, which is exponentially large to the problem instance in the combinatorial setting.

To the best of our knowledge, there exists no polynomial-time algorithm for CPE-PL and there exists no polynomial-time adaptive algorithm for CPE-BL with general constraints in the literature. It is worth to mention that, for CPE-BL in the Top-k setting, there is a naive reduction to the classic CPE-MB, since an unbiased estimate can be obtained for the difference between the two base arms by comparing two k-base arm queries with one base arm difference. For more general combinatorial constraints, however, there are no simple reductions (see Appendix C for details). Our work provides the first learnability result for CPE-PL and the first adaptive algorithm for CPE-BL with general combinatorial constraints. Our contributions can be summarized as follows:

- We propose the novel model of CPE-PL, which simultaneously addresses limited feedback, general reward function, and combinatorial action space.
- We present the first polynomial-time algorithm for CPE-PL (Algorithm 1) and analyze the sample complexity bound (Theorem 1).
- We further design the first polynomial-time adaptive algorithm for CPE-BL under general combinatorial constraints (Algorithm 5) whose sample complexity has lighter dependence of $\Delta_{\min}$ (Theorem 2).

Comparison between our results and existing work in the fixed confidence setting is summarized in Table 1.\(^1\)

2 Problem Statements

Combinatorial pure exploration with partial-monitoring linear feedback (CPE-PL). In the CPE-PL problem, we have $d$ base arms numbered $1, 2, \ldots, d$. We define $\mathcal{X} \subseteq \{0, 1\}^d$ as a set of all super arms satisfying a certain combinatorial structures such as size-$k$, matroids, paths, and matchings. Let $m$ denote the maximum number of base arms that a super arm in $\mathcal{X}$ contains, i.e. $m = \max_{x \in \mathcal{X}} \|x\|_1$, ($m \leq d$). For each super arm $x \in \mathcal{X}$, there is a transformation matrix $M_x \in \mathbb{R}^{m \times d}$, whose row dimension $m_x$ depends on $x$. There is an unknown environment vector $\theta \in \mathbb{R}^d$ with $\|\theta\|_2 \leq L$. At each timestep $t$, a learner pulls a super arm $x_t$ and observes a random linear feedback vector $y_t = M_{x_t}(\theta + \eta_t) \in \mathbb{R}^{m \times 1}$, where $\eta_t$ is a zero-mean noise vector bounded in $[-1, 1]^d$ and it is independent among different timestep $t$. Meanwhile, the learner gains a random reward with expectation of $\bar{r}(x_t, \theta)$. Note that for each pull of super arm $x_t$, the actual expected reward $\bar{r}(x_t, \theta)$ may not be part of the linear feedback vector $y_t$ and thus may not be directly observed by the learner, and in the case of pure exploration, the reward obtained in each round is irrelevant since the objective is not on cumulative reward or cumulative regret. Let $x^* = \arg\max_{x \in \mathcal{X}} \bar{r}(x, \theta)$ denote the optimal super arm with the maximum expected reward, and $\Delta_t$ denote the gap of expected rewards between $x^*$ and the super arm with the $i$-th largest expected reward. Given a confidence parameter

\(^1\)Recently, Kuroki et al. [33] studied a variant of CPE, where the offline optimization is the densest subgraph problem and the learner observes full-bandit feedback for a set of edges. Their algorithms and analysis use the property of the average degree, and thus they cannot be directly applied to solve either CPE-PL or CPE-BL.

\(^2\)Notations appearing in the table but not relevant in our problem setting are given below: $\rho(\lambda) = \max_{x \in \mathcal{X}} \|x\|_{\tilde{M}(\lambda)^{-1}}$, $\tilde{\Delta}_i = \theta_i - \theta_{i+1}$ if $i \leq k$ and $\theta_k - \theta_i$ otherwise. $H_x = \max_{x \subseteq x_j \subseteq \mathcal{X}} \frac{\bar{r}_x(x, x_j)}{\max(\Delta_i^2 \Delta_j^2)}$ where $\Delta = (x^* - x_1)^\top \theta$ if $x_1 \neq x^*$, $\arg\min_{x \in \mathcal{X}} x^* - x$ otherwise, and $\bar{r}_x(x, x_j)$ is a term defined by the optimal solution to a convex optimization (see (11) in [44]). $S_t = \{x \in \mathcal{X} : (x^* - x)^\top \theta \leq 4 \cdot 2^{-t}\}$. $\mathcal{Y}(S) = \{x - x' : \forall x, x' \in \mathcal{X}, x \neq x'\}$. $\tilde{\rho}(\mathcal{Y}(S)) = \min_{\lambda \in \Delta(\mathcal{X})} \max_{x \in \mathcal{Y}(S)} \|x\|_{\tilde{M}(\lambda)^{-1}}$. 

3
δ ∈ (0, 1), the objective of the learner is to use few samples as possible to identify the optimal super arm x∗ with probability at least 1 − δ. This is often called the fixed confidence setting in the bandit literature, and the number of samples required by the learner is called the sample complexity.

The regret minimization scheme of the CPE-PL problem has been studied in [37]. In this paper, we study the pure exploration scheme and inherit two technical assumptions from [37] in order to design an efficient pure exploration algorithm.

**Assumption 1** (Lipschitz continuity of the expected reward function). There exists a constant Lp such that for any x ∈ X and any θ1, θ2 ∈ Rd, |r(x, θ1) − r(x, θ2)| ≤ Lp∥θ1 − θ2∥2.

**Assumption 2** (Global observer set). There exists a global observer set Σ = {x1, x2, . . . , x|Σ|} ⊆ X, such that the stacked \( \sum_{i=1}^{|\Sigma|} m_{x_i} \)-by-d transformation matrix \( M_\Sigma = (M_{x_1}; M_{x_2}; . . . ; M_{x_{|\Sigma|}}) \) is of full column rank, i.e., rank(\( M_\Sigma \)) = d. Then, the Moore-Penrose pseudoinverse \( M_\Sigma^+ \) satisfies \( M_\Sigma^+ M_\Sigma = I_d \), where \( I_d \) is the d-by-d identity matrix.

**Combinatorial pure exploration with full-bandit linear feedback (CPE-BL).** CPE-BL is an important problem in the CPE-PL problem, for which the learner observes full-bandit linear feedback, i.e. \( M_x = x^\top \) and gains linear expected reward, i.e. \( r(x, \theta) = x^\top \theta \) after each play. The CPE-BL problem is the combinatorial adaptation of the best arm identification problem in linear bandits, where the action space \( X \subseteq \mathbb{R}^d \) but \( |X| \) is often assumed to be small. The combinatorial action space setting in the CPE-BL problem better suits for the large structured action sets such as size-k, matroids, paths, and matchings, and requests additional techniques to address the computational challenges.

For clarity, we also introduce the following notations. Let \( |d| = \{1, 2, . . . , d\} \). For a vector \( x \in \mathbb{R}^d \) and a matrix \( B \in \mathbb{R}^{d \times d} \), let \( \|x\|_B = \sqrt{x^\top B x} \). For a positive definite matrix \( B \in \mathbb{R}^{d \times d} \), we use \( B^{1/2} \) to denote the unique positive definite \( d \times d \) matrix whose square is \( B \). For a given family \( X \), we use \( \triangle(X) \) to denote the set of probability distributions over a family \( X \). For a distribution \( \lambda \in \triangle(X) \), we use \( \text{supp}(\lambda) \) to denote its support, i.e., \( \text{supp}(\lambda) = \{x : \lambda(x) > 0\} \). We define \( M(\lambda) = \mathbb{E}_{z \sim \lambda}[zz^\top] \).

### Table 1: Comparison between our results and existing results for CPE-PL (BL)

| Algorithm       | Sample complexity | Case          | Reward Feedback | Strategy | Time |
|-----------------|-------------------|---------------|-----------------|----------|------|
| GCB-PE (Thm. 1) | \( O(\frac{d^3 \log \sum_{i=1}^d \frac{1}{\lambda_i^2})}{\sum_{i=1}^d \frac{1}{\lambda_i^2})} \) | General | Nonlinear PL | Static | Poly(d) |
| PolyALBA (Thm. 2) | \( \tilde{O} (\frac{\log \sum_{i=1}^d \frac{1}{\lambda_i^2}}{\lambda_i^2} + d \log \frac{\log \sum_{i=1}^d \frac{1}{\lambda_i^2}}{\sum_{i=1}^d \frac{1}{\lambda_i^2}}) \) | General | Linear PL | Adaptive | Poly(d) |
| ICB [34] | \( \tilde{O} (\frac{d^3 \log \sum_{i=1}^d \frac{1}{\lambda_i^2}}{\sum_{i=1}^d \frac{1}{\lambda_i^2}}) \) | General | Linear | Adaptive | Poly(d) |
| SAQ [34] | \( \tilde{O} (\frac{\log \sum_{i=1}^d \frac{1}{\lambda_i^2}}{\lambda_i^2} + d \log \frac{\log \sum_{i=1}^d \frac{1}{\lambda_i^2}}{\lambda_i^2} + d \log \frac{\log \sum_{i=1}^d \frac{1}{\lambda_i^2}}{\lambda_i^2}) \) | Top-k | Linear | Static | Poly(d) |
| CSAR [49] | \( \tilde{O} (\frac{d^3 \sum_{i=1}^d \frac{1}{\lambda_i^2}}{\lambda_i^2}) \) | Top-k | Adaptive | Poly(d) |
| \( X^\text{Y}-\text{static} [42] \) | \( \tilde{O} (\frac{d^3 \sum_{i=1}^d \frac{1}{\lambda_i^2}}{\lambda_i^2} + d \log \frac{\log \sum_{i=1}^d \frac{1}{\lambda_i^2}}{\sum_{i=1}^d \frac{1}{\lambda_i^2}}) \) | \( X \subseteq \mathbb{R}^d \) | Linear | Static | \( \Omega(|X|) \) |
| Explore-Verify [28] | \( \tilde{O} (\frac{d^3 \sum_{i=1}^d \frac{1}{\lambda_i^2}}{\lambda_i^2} + d \log \frac{\log \sum_{i=1}^d \frac{1}{\lambda_i^2}}{\sum_{i=1}^d \frac{1}{\lambda_i^2}}) \) | \( X \subseteq \mathbb{R}^d \) | Linear | Static | \( \Omega(|X|) \) |
| LinGapE [44] | \( \tilde{O} (\frac{d^3 \sum_{i=1}^d \frac{1}{\lambda_i^2}}{\lambda_i^2} + d \log \frac{\log \sum_{i=1}^d \frac{1}{\lambda_i^2}}{\sum_{i=1}^d \frac{1}{\lambda_i^2}}) \) | \( X \subseteq \mathbb{R}^d \) | Linear | Static | \( \Omega(|X|) \) |
| J- ElimTil [43] | \( \tilde{O} (\frac{d^3 \sum_{i=1}^d \frac{1}{\lambda_i^2}}{\lambda_i^2} + d \log \frac{\log \sum_{i=1}^d \frac{1}{\lambda_i^2}}{\sum_{i=1}^d \frac{1}{\lambda_i^2}}) \) | \( X \subseteq \mathbb{R}^d \) | Linear | Static | \( \Omega(|X|) \) |
| ALBA [43] | \( \tilde{O} (\frac{d^3 \sum_{i=1}^d \frac{1}{\lambda_i^2}}{\lambda_i^2} + d \log \frac{\log \sum_{i=1}^d \frac{1}{\lambda_i^2}}{\sum_{i=1}^d \frac{1}{\lambda_i^2}}) \) | \( X \subseteq \mathbb{R}^d \) | Linear | Static | \( \Omega(|X|) \) |
| RAGE [18] | \( \tilde{O} (\frac{d^3 \sum_{i=1}^d \frac{1}{\lambda_i^2}}{\lambda_i^2} + d \log \frac{\log \sum_{i=1}^d \frac{1}{\lambda_i^2}}{\sum_{i=1}^d \frac{1}{\lambda_i^2}}) \) | \( X \subseteq \mathbb{R}^d \) | Linear | Static | \( \Omega(|X|) \) |
| Lower Bound [18] | \( \min_{\lambda \in X} \lambda^\top \min_{x \in X} \lambda \) | \( X \subseteq \mathbb{R}^d \) | Linear | Static | - |

\[ \frac{\log \sum_{i=1}^d \frac{1}{\lambda_i^2}}{\lambda_i^2} + d \log \frac{\log \sum_{i=1}^d \frac{1}{\lambda_i^2}}{\sum_{i=1}^d \frac{1}{\lambda_i^2}} \]
and $\hat{M}(\lambda) = (\sum_{x \in supp(\lambda)} xx^\top)$. We denote the maximum (minimal) eigenvalue of matrix $B$ by $\xi_{\text{max}}(B)(\xi_{\text{min}}(B))$.

## 3 Algorithm for CPE-PL

In this section, we present a polynomial-time algorithm for CPE-PL, namely GCB-PE, which is based on the GCB algorithm of [37], and provide the sample complexity bound.

We illustrate GCB-PE in Algorithm 1. GCB-PE estimates the environment vector $\theta$ by repeatedly pulling the global observer set $\sigma = \{x_1, x_2, \ldots, x_{|\sigma|}\}$, which in turn helps estimate the expected rewards $\bar{r}(x, \theta)$ of all super arms $x \in \mathcal{X}$ using the Lipschitz continuity (Assumption 1). We call one pull of global observer set $\sigma$ one exploration round, the specific procedure of which is described as follows: for the $n$-th exploration round, the learner plays all actions in $\sigma = \{x_1, x_2, \ldots, x_{|\sigma|}\}$ once and respectively observes feedback $y_1, y_2, \ldots, y_{|\sigma|}$, the stacked vector of which is denoted by $\bar{y}_n = (y_1; y_2; \ldots; y_{|\sigma|})$. The estimate of environment vector $\theta$ in this exploration round is $\hat{\theta}_n = M_+^\top \bar{y}_n$, where $M_+^\top$ is the Moore-Penrose pseudoinverse of $M_\sigma$. From Assumption 2, we have $E[\hat{\theta}_n] = \theta$. Then, we can use the independent estimates in multiple rounds, i.e., $\hat{\theta}(n) = \frac{1}{n} \sum_{j=1}^n \hat{\theta}_j$, to obtain an accurate estimate of $\theta$.

Similar to [37], we define a constant $\beta_{\sigma} := \max_{m_1, \ldots, m_{|\sigma|}} \|y_{m_1} \cdots y_{m_{|\sigma|}}\|_2$, which only depends on global observer set $\sigma$. $\beta_{\sigma}$ bounds the estimate error of one exploration round, i.e., for any $n$,

$$\|\hat{\theta}_n - \theta\|_2 \leq \beta_{\sigma}. \quad (1)$$

The proof of Eq. (1) is given in Appendix B.1. Based on $\beta_{\sigma}$, we further design a global confidence radius $\text{rad}_n = \sqrt{\frac{2\beta_{\sigma}^2 |\sigma|^2 \log \left( \frac{4e^2 |\sigma|^2}{\delta} \right)}{n}}$ for the estimate $\hat{\theta}(n)$, and show that with high probability, $\text{rad}_n$ bounds the estimate error of $\hat{\theta}(n)$ (see Lemma 2 in Appendix B.2).

Comparing with GCB in [37], which works for the regret minimization metric of the combinatorial partial monitoring game with linear feedback problem, GCB-PE targets the best action identification and mainly controls the stopping time of the exploration phase rather than balancing the frequency of exploration and exploitation phases. For the pure exploration metric, our global confidence radius $\text{rad}_n$ is designed to bound the estimate error. In addition, the stopping condition, which uses the designed confidence radius and Lipschitz continuity of the expected reward function, is also newly adopted to fit the CPE-PL setting.

The computational efficiency of GCB-PE relies on the polynomial-time offline maximization oracle for the specific combinatorial instance, which is used in the two argmax operations in GCB-PE. It is reasonable to assume the existence of polynomial-time offline maximization oracle, otherwise we cannot efficiently address the exponentially large action space even if the real environment vector $\theta$ is known.

GCB-PE provides a general algorithmic framework to simultaneously address the partial linear feedback, general expected reward and combinatorial action space. Theorem 1 shows the sample complexity upper bound for GCB-PE. To our best knowledge, it gives the first theoretical result for the CPE-PL problem.

**Theorem 1.** With probability at least $1 - \delta$, the GCB-PE algorithm (Algorithm 1) will return the optimal super arm $x^*$ with sample complexity

$$O \left( \frac{|\sigma| \beta_{\sigma}^2 L_p^2}{\Delta_{\text{min}}^2} \log \left( \frac{\beta_{\sigma}^2 L_p^2}{\Delta_{\text{min}}^2 \delta} \right) \right).$$

**Corollary 1.** Consider the linear expected reward case in which $\bar{r}(x, \theta) = x^\top \theta$. Let $m$ ($m \leq d$) denote the maximum number of base arms a super arm contains. With probability at least $1 - \delta$, the
Algorithm 1: GCB-PE

Input: Confidence level $\delta$, global observer set $\sigma$, constant $\beta$,
Lipschitz constant $L_p$

for $s = 1, \ldots, |\sigma|$ do
    Pull $x_s$ in observer set $\sigma$, and observe $y_s$;
    $n \leftarrow 1$;
    $\hat{y}_1 = (y_1; y_2; \ldots; y_{|\sigma|})$;
    $\hat{\theta}_1 = M_{\sigma}^+ \hat{y}_1$ and $\hat{\theta}(1) = \hat{\theta}_1$;
    while true do
        $\hat{x} = \arg\max_{x \in X} \tilde{r}(x, \hat{\theta}(n))$;
        $\hat{x}^- = \arg\max_{x \in X \setminus \{\hat{x}\}} \tilde{r}(x, \hat{\theta}(n))$;
        rad$_n \leftarrow \sqrt{\frac{2\beta^2 \log(4n^2e^2)}{n}}$;
        if $\tilde{r}(\hat{x}, \hat{\theta}(n)) - \tilde{r}(\hat{x}^-, \hat{\theta}(n)) > 2L_p \cdot \text{rad}_n$ then
            return $\hat{x}$;
        else
            for $s = 1, \ldots, |\sigma|$ do
                Pull $x_s$ in observer set $\sigma$, and observe $y_s$;
                $n \leftarrow n + 1$;
                $\hat{y}_n = (y_1; y_2; \ldots; y_{|\sigma|})$;
                $\hat{\theta}_n = M_{\sigma}^+ \hat{y}_n$;
                $\hat{\theta}(n) = \frac{1}{n} \sum_{j=1}^{n} \hat{\theta}_j$;
            end
        end
    end
end

Output: $\hat{x}$

GCB-PE algorithm (Algorithm 1) will return the optimal super arm $x^*$ with sample complexity

$$O\left(\frac{|\sigma| \beta^2 m^2}{\Delta_{\min}^2 \log \left(\frac{\beta^2 m^2}{\Delta_{\min}^2 \delta}\right)}\right).$$

We remark that $\beta = \text{Poly}(d)$ for several practical cases. In the scenario of task assignments in crowdsourcing where the combinatorial action space is characterized by matchings [37], after each pull, we may be able to observe the reward of a single edge (pair) rather than all edges in the pulled matching, i.e., $M_x$ may contain a single row with all entries “0” except one “1”. In this case, we can choose the global observer set $\sigma$ such that $M_{\sigma}$ is an identity matrix. Then, we obtain $\beta_{\sigma} = \sqrt{d}$. In the scenario of online ranking with feedback at top-ranked item [9], where $\theta \in \{0, 1\}^d$ and $X = d!$ permutations and each action $x$ has $M_x \in \{0, 1\}^d$ with “1” in the place of the item which is ranked at the top by $x$ and “0” everywhere else. In this case, we can select the global observer set $\sigma$ as a set of any $d$ actions such that $M_{\sigma}$ is an identity matrix, and again we have $\beta_{\sigma} = \sqrt{d}$.

4 Algorithm for CPE-BL

The main focus of this section is to devise an adaptive algorithm for CPE-BL, the subproblem of CPE-PL. We propose a polynomial-time adaptive algorithm, namely PolyALBA, and provide the sample complexity bound that has lighter dependence of minimum gap $\Delta_{\min}$. Algorithm 5 details the procedure of PolyALBA.

ALBA algorithm [43]. Before stating the main algorithm, we introduce the Adaptive Linear Best Arm algorithm (ALBA) for BAI-LB [43] (see Algorithm 2 for its description), which is the key subroutine of our proposed method PolyALBA. First, we describe the randomized least-square
illustrates the procedure. ALBA is an elimination-based algorithm, where in round \( t \) it identifies the top \( \lambda_t \) arms and discards the remaining arms by means of \( \text{ElimTil}_p \) (Algorithm 4). Note that if we naively perform ALBA for \( \mathcal{X} \), ALBA takes \( \Omega(|\mathcal{X}|) \) time, which is infeasible to the problem instance in the combinatorial setting.

**Algorithm 2: ALBA(S, δ) [43]**

**Algorithm 3: VectorEst(λ, n)**

**Main algorithm.** Now we describe the details of our proposed algorithm PolyALBA, in which ALBA is invoked but it runs in polynomial time. To reduce the computational cost of ALBA, PolyALBA has the novel preparation procedure before invoking ALBA; we design an efficient sampling scheme in the first epoch \( q = 0 \) for identifying a set of \( d \) super arms with the highest empirical means. In this preparation sampling scheme, we employ a static allocation strategy with a fixed distribution \( \lambda \in \Delta(\mathcal{X}) \) which has the polynomial-size support. Algorithm 6 illustrates the procedure of computing one distribution \( \lambda \in \Delta(\mathcal{X}) \), and derives the key parameter \( \alpha \) for the first epoch in PolyALBA, on which we will show the good property later. At each round \( r \), we obtain the estimate \( \hat{\theta}_r \) by VectorEst(\( \lambda, n \)) where the number of samples \( n \) are carefully set. Then, we compute the empirical best \( d + 1 \) super arms with respect to the estimate \( \hat{\theta}_r \). Note that this procedure can be done in polynomial time due to the method by Lawler [36], called the Lawler’s \( k \)-best procedure, as long as max_{\( x \in \mathcal{X} \)} \( \hat{\theta}_r^\top x \) is solved in polynomial time. The first epoch ends with a set \( S_1 = \{ \hat{x}_1, \ldots, \hat{x}_d \} \). Then PolyALBA focuses on sampling a set of seemingly near-optimal super-arms \( S_1 \) by means of ALBA and returns \( \hat{x}_* \) as its output.

**Theoretical analysis.** We present the theoretical analysis of PolyALBA, and briefly explain how to achieve both polynomial-time complexity and optimality. Now we provide a problem-dependent
Algorithm 5: PolyALBA

Input: $d$-base arms, confidence level $\delta$, $c_0 = \max\{4L^2, 3\}$.

Compute a distribution $\lambda \leftarrow \lambda_{X_a}^*$ and parameter $\alpha \leftarrow \sqrt{md/\xi_{min}(M(\lambda_{X_a}^*))}$ by Algorithm 6;

Set $q \leftarrow 0$ and $\delta_q \leftarrow \frac{\alpha}{8} \left(\frac{d}{q+1}\right)^2$:

$r \leftarrow 1$;

while true do

Set $\varepsilon_r \leftarrow \frac{1}{r}$ and $\delta_r \leftarrow \frac{6 \delta_q}{r}$;

$\ell(\varepsilon) = \frac{24 \sqrt{md+4d^2+d+\alpha^2d}}{\varepsilon^2}$;

$\hat{\theta}_r \leftarrow \text{VectorEst}(\lambda, c_0 \ell(\frac{\varepsilon}{2}) \ln(\frac{5|X|}{\delta_r}))$;

Select $d+1$ super arms $\hat{x}_1, \ldots, \hat{x}_d, \hat{x}_{d+1}$ with the highest $d+1$ empirical means $x^\top \hat{\theta}_r$ in all $x \in X$;

if $\hat{x}_1^\top \hat{\theta}_r - \hat{x}_{d+1}^\top \hat{\theta}_r > \varepsilon_r$ then

$\hat{x}_1 \leftarrow \{\hat{x}_1, \ldots, \hat{x}_d\}$

break;

$r \leftarrow r + 1$;

$q \leftarrow 1$;

$\hat{x}^* \leftarrow$ output by ALBA$(S_q, \delta_q)$

Output: $\hat{x}^*$}

Algorithm 6: Computing a distribution $\lambda$

Input: $d$-base arms

Choose any $d$ super arms $X_a = \{b_1, \ldots, b_d\}$ from $X$, such that $\text{rank}(X) = d$ where $X = \{b_1, \ldots, b_d\}$;

$\lambda_{X_a}^* \leftarrow \arg\min_{\lambda \in \Delta(X_a)} \max_{x \in X_a} x^\top M(\lambda)^{-1} x$ by Algorithm 8 in Appendix D;

$\alpha \leftarrow \sqrt{md/\xi_{min}(M(\lambda_{X_a}^*))}$;

Output: $\lambda_{X_a}^*$ and $\alpha$

Theorem 2. With probability at least $1 - \delta$, the PolyALBA algorithm (Algorithm 5) will return the best super arm $x^*$ with sample complexity

$$O\left(\frac{c_0 d (\alpha \sqrt{m} + \alpha^2)}{\Delta_{d+1}^2} \left(\ln \delta^{-1} + \ln |X| + \ln \ln \Delta_{d+1}^{-1}\right) + \sum_{i=2}^{d} \frac{c_0}{\Delta_i^2} (\ln \delta^{-1} + \ln |X| + \ln \ln \Delta_i^{-1})\right).$$

The first term in Theorem 2 is for the preparation procedure in the first epoch and the second term is for the remaining epochs required by subroutine ALBA. As shown in Theorem 2, our sample complexity bound has lighter dependence of $1/\Delta_{min}^2$, compared with the existing result (see Table 1). Now we explain the key role for the polynomial-time complexity of PolyALBA in the first epoch played by the distribution $\lambda_{X_a}^*$ and parameter $\alpha$. Notice that even if we employ a uniform distribution on a polynomial-size support $X_a \subseteq X$, i.e., $\lambda_{X_a} = (1/|X_a|)_{x \in X_a}$, computing the maximal confidence bound $\max_{x \in X} \|x\|_{M(\lambda_{X_a})^{-1}}$ is NP-hard, while many (UCB-based) algorithms in LB simply used a brute force. Despite computational challenges, by utilizing a property of G-optimal design [38], PolyALBA runs in polynomial time while guaranteeing the optimality. In the following lemma, we show that $\alpha \sqrt{d}$ gives the upper bound on the maximal ellipsoidal norm associated to $M(\lambda_{X_a})^{-1}$.

Lemma 1. For $\lambda_{X_a}^*$ and $\alpha$ obtained by Algorithm 6, it holds that $\max_{x \in X} \|x\|_{M(\lambda_{X_a})^{-1}} \leq \alpha \sqrt{d}$. 

8
From the equivalence theorem for optimal experimental designs (Proposition 1 in Appendix A), it holds that
\[ \min_{\lambda \in \Delta(\mathcal{X})} \max_{x \in \mathcal{X}} \|x\|_{M(\lambda)^{-1}} = \sqrt{d}. \]
From this fact and Lemma 1, we see that \( \lambda^*_x \) is \( \alpha (\geq 1) \)-approximate solution to \( \min_{\lambda \in \Delta(\mathcal{X})} \max_{x \in \mathcal{X}} \|x\|_{M(\lambda)^{-1}} \) where \( \mathcal{X} \) can be defined by general combinatorial constraints. Note that \( \alpha \) can be easily obtained by computing \( \xi_{\min}(\tilde{M}(\lambda^*_x)) \) (recall that \( \tilde{M}(\lambda) = \sum_{x \in \text{supp}(\lambda)} xx^T \)). Therefore, by employing \( \lambda^*_x \) and a prior knowledge of its approximation ratio \( \alpha \), we can guarantee that the preparation sampling scheme identifies a set \( S_1 \) containing the optimal super arm \( x^* \) with high probability. In the remaining epochs, PolyALBA can successfully focus on sampling near-optimal super arms by ALBA owing to the optimality of \( S_1 \).

Finally, we note the further improvement. If we compute \( \min_{\lambda \in \Delta(\mathcal{X})} \max_{x \in \mathcal{X}} \|x\|_{M(\lambda)^{-1}} \) exactly, we have \( \alpha = 1 \). If we approximately solve it, \( \alpha \) is independent on the arm-selection ratio but it can depend on the support of \( \lambda \). Lemma 1 indicates that choosing \( \text{supp}(\lambda) \) that maximizes \( \xi_{\min}(\tilde{M}(\lambda)) \) gives the better bound. Such a design is so called the \( E \)-optimal design, i.e., the goal is to minimize the maximum eigenvalue of the error covariance [38]. Note that if we are allowed to pull unit vectors, we have \( \xi_{\min}(\tilde{M}(\lambda)) \geq 1 \). For general cases, more sophisticated algorithm by the Ellipsoid method [22] is given in Appendix D as one alternative of Algorithm 6. With some additional assumptions, this approach provides the better choice of \( \lambda \) (or equivalently \( \alpha \)) than Algorithm 6 in terms of the expectation value.

5 Conclusion and Future Work

We introduce a novel problem, combinatorial pure exploration with partial linear feedback (CPE-PL), which simultaneously models limited feedback, general (possibly nonlinear) reward and combinatorial action space, and finds various applications such as recommendation systems and crowdsourcing. We also study the important subproblem of CPE-PL, i.e., CPE with full-bandit feedback (CPE-BL), in which both the feedback and expected reward are linear. For CPE-PL, we propose a general polynomial-time algorithmic framework \( \text{GCB-PE} \) with sample complexity analysis. For CPE-BL, we further design a dedicated algorithm PolyALBA with better sample complexity. Our algorithms and analysis provide an efficient solution to identifying the optimal action under combinatorial action space and partial feedback.

There are several interesting directions worth further investigation. First, it is open to prove a lower bound of polynomial-time algorithms for both CPE-PL and CPE-BL. Another challenging direction is to design efficient algorithms for specific combinatorial cases to choose the global observer set \( \sigma \) and the distribution \( \lambda^*_x \), and derive specific sample complexity bounds. Furthermore, the extension of CPE-PL to nonlinear feedback is also a practical and valuable problem.

Acknowledgement

Yuko Kuroki is partially supported by Microsoft Research Asia and KAKENHI 18J23034.
References

[1] Y. Abbasi-Yadkori, D. Pál, and C. Szepesvári. Improved algorithms for linear stochastic bandits. In Proc. NIPS’14, pages 2312–2320, 2011.

[2] J.-Y. Audibert, S. Bubeck, and R. Munos. Best arm identification in multi-armed bandits. In Proc. COLT’10, pages 41–53, 2010.

[3] S. Bubeck, T. Wang, and N. Viswanathan. Multiple identifications in multi-armed bandits. In Proc. ICML’13, pages 258–265, 2013.

[4] T. Cao and A. Krishnamurthy. Disagreement-based combinatorial pure exploration: Sample complexity bounds and an efficient algorithm. In Proc. COLT’19, pages 558–588, 2019.

[5] W. Cao, J. Li, Y. Tao, and Z. Li. On top-k selection in multi-armed bandits and hidden bipartite graphs. In Proc. NIPS’15, pages 1036–1044, 2015.

[6] A. Carpentier and A. Locatelli. Tight (lower) bounds for the fixed budget best arm identification bandit problem. In Proc. COLT’16, pages 590–604, 2016.

[7] A. R. Chaudhuri and S. Kalyanakrishnan. PAC identification of many good arms in stochastic multi-armed bandits. In Proc. ICML’19, pages 991–1000, 2019.

[8] S. Chaudhuri and A. Tewari. Online ranking with top-1 feedback. In Proc. AISTATS’15, pages 129–137, 2015.

[9] S. Chaudhuri and A. Tewari. Phased exploration with greedy exploitation in stochastic combinatorial partial monitoring games. In Proc. NIPS’16, pages 2433–2441, 2016.

[10] S. Chaudhuri and A. Tewari. Online learning to rank with top-k feedback. The Journal of Machine Learning Research, 18(1):3599–3648, 2017.

[11] L. Chen, A. Gupta, and J. Li. Pure exploration of multi-armed bandit under matroid constraints. In Proc. COLT’16, pages 647–669, 2016.

[12] L. Chen, A. Gupta, J. Li, M. Qiao, and R. Wang. Nearly optimal sampling algorithms for combinatorial pure exploration. In Proc. COLT’17, pages 482–534, 2017.

[13] L. Chen and J. Li. On the optimal sample complexity for best arm identification. arXiv preprint, arXiv:1511.03774, 2015.

[14] S. Chen, T. Lin, I. King, M. R. Lyu, and W. Chen. Combinatorial pure exploration of multi-armed bandits. In Proc. NIPS’14, pages 379–387, 2014.

[15] V. Dani, T. P. Hayes, and S. M. Kakade. Stochastic linear optimization under bandit feedback. In Proc. COLT’08, pages 355–366, 2008.

[16] E. Even-Dar, S. Mannor, and Y. Mansour. PAC bounds for multi-armed bandit and Markov decision processes. In Proc. COLT’02, pages 255–270, 2002.

[17] E. Even-Dar, S. Mannor, and Y. Mansour. Action elimination and stopping conditions for the multi-armed bandit and reinforcement learning problems. Journal of Machine Learning Research, 7:1079–1105, 2006.

[18] T. Fiez, L. Jain, K. G. Jamieson, and L. Ratliff. Sequential experimental design for transductive linear bandits. In Proc. NeurIPS’19, pages 10667–10677, 2019.

[19] V. Gabillon, M. Ghavamzadeh, and A. Lazaric. Best arm identification: A unified approach to fixed budget and fixed confidence. In Proc. NIPS’12, pages 3212–3220, 2012.
[20] V. Gabillon, M. Ghavamzadeh, A. Lazaric, and S. Bubeck. Multi-bandit best arm identification. In Proc. NIPS’11, pages 2222–2230, 2011.

[21] V. Gabillon, A. Lazaric, M. Ghavamzadeh, R. Ortner, and P. Bartlett. Improved learning complexity in combinatorial pure exploration bandits. In Proc. AISTATS’16, pages 1004–1012, 2016.

[22] M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. Combinatorica, 1:169–197, 1981.

[23] S. Huang, X. Liu, and Z. Ding. Opportunistic spectrum access in cognitive radio networks. In Proc. INFOCOM’08, pages 1427–1435, 2008.

[24] K. Jamieson, M. Malloy, R. Nowak, and S. Bubeck. lil’UCB: An optimal exploration algorithm for multi-armed bandits. In Proc. COLT’14, pages 423–439, 2014.

[25] S. Kalyanakrishnan and P. Stone. Efficient selection of multiple bandit arms: Theory and practice. In Proc. ICML’10, pages 511–518, 2010.

[26] S. Kalyanakrishnan, A. Tewari, P. Auer, and P. Stone. PAC subset selection in stochastic multi-armed bandits. In Proc. ICML’12, pages 655–662, 2012.

[27] Z. Karnin, T. Koren, and O. Somekh. Almost optimal exploration in multi-armed bandits. In Proc. ICML’13, pages 1238–1246, 2013.

[28] Z. S. Karnin. Verification based solution for structured mab problems. In Proc. NIPS’16, pages 145–153, 2016.

[29] E. Kaufmann, O. Cappé, and A. Garivier. On the complexity of best-arm identification in multi-armed bandit models. Journal of Machine Learning Research, 17:1–42, 2016.

[30] E. Kaufmann and S. Kalyanakrishnan. Information complexity in bandit subset selection. In Proc. COLT’13, pages 228–251, 2013.

[31] Y. Kawase and H. Sumita. Randomized strategies for robust combinatorial optimization. In Proc. AAAI ’19, pages 7876–7883, 2019.

[32] J. Kiefer and J. Wolfowitz. The equivalence of two extremum problems. Canadian Journal of Mathematics, 12:363–366, 1960.

[33] Y. Kuroki, A. Miyauuchi, J. Honda, and M. Sugiyama. Online dense subgraph discovery via blurred-graph feedback. In Proc. ICML’20, accepted, 2020.

[34] Y. Kuroki, L. Xu, A. Miyauuchi, J. Honda, and M. Sugiyama. Polynomial-time algorithms for multiple-arm identification with full-bandit feedback. Neural Computation, in press, 2020.

[35] T. Lattimore and C. Szepesvari. The End of Optimism? An Asymptotic Analysis of Finite-Armed Linear Bandits. In Proc. AISTATS’17, pages 728–737, 2017.

[36] E. L. Lawler. A procedure for computing the k best solutions to discrete optimization problems and its application to the shortest path problem. Management Science, 18(7):401–405, 1972.

[37] T. Lin, B. Abraham, R. Kleinberg, J. Lui, and W. Chen. Combinatorial partial monitoring game with linear feedback and its applications. In Proc. ICML’14, pages 901–909, 2014.

[38] F. Pukelsheim. Optimal Design of Experiments. Society for Industrial and Applied Mathematics, 2006.
[39] I. Rejwan and Y. Mansour. Top-k combinatorial bandits with full-bandit feedback. In Proc. ALT’ 20, pages 752–776, 2020.

[40] A. Roy Chaudhuri and S. Kalyanakrishnan. PAC identification of a bandit arm relative to a reward quantile. In Proc. AAAI’ 17, pages 1977–1985, 2017.

[41] P. Rusmevichientong and D. P. Williamson. An adaptive algorithm for selecting profitable keywords for search-based advertising services. In Proc. EC ’06, pages 260–269, 2006.

[42] M. Soare, A. Lazaric, and R. Munos. Best-arm identification in linear bandits. In Proc. NIPS’14, pages 828–836, 2014.

[43] C. Tao, S. Blanco, and Y. Zhou. Best arm identification in linear bandits with linear dimension dependency. In Proc. ICML’18, pages 4877–4886, 2018.

[44] L. Xu, J. Honda, and M. Sugiyama. A fully adaptive algorithm for pure exploration in linear bandits. In Proc. AISTATS’18, pages 843–851, 2018.

[45] Y. Zhou, X. Chen, and J. Li. Optimal PAC multiple arm identification with applications to crowdsourcing. In Proc. ICML’14, pages 217–225, 2014.
Appendix

A Equivalence theorem for optimal experimental design

We introduce the following equivalence theorem in [32] adopted our setting of CPE-BL, which will be used in our analysis.

**Proposition 1** (Kiefer and Wolfowitz [32]). Define $M(\lambda) = \mathbb{E}_{z \sim \lambda}[zz^T]$ for any distribution $\lambda$ supported on $X \subseteq \mathbb{R}^d$. We consider two extremum problems.

The first is to choose $\lambda$ so that

\[ (1) \lambda \text{ maximizes } \det M(\lambda) \quad (D\text{-optimal design}) \]

The second one is to choose $\lambda$ so that

\[ (2) \lambda \text{ minimizes } \max_{x \in X} x^T M(\lambda)^{-1} x \quad (G\text{-optimal design}) \]

We note that $\mathbb{E}_{x \sim \lambda}[x^T M(\lambda)^{-1} x]$ is $d$, hence, $\max_{x \in X} x^T M(\lambda)^{-1} x \geq d$, and thus a sufficient condition for $\lambda$ to satisfy (2) is

\[ (3) \max_{x \in X} x^T M(\lambda)^{-1} x = d. \]

Statements (1), (2) and (3) are equivalent.

B Missing proofs

B.1 Proof of Equation 1

**Proof.** We prove Equation 1 using similar techniques in [37].

Recall that in the GCB-PE algorithm (Algorithm 1), $\hat{\theta}_n$ is the estimate of the environment vector $\theta$ in the $n$-th exploration round. For any $n$,

\[
\|\hat{\theta}_n - \theta\|_2 \\
= \|M_\sigma^+] y_n - M_\sigma^+] M_\sigma \theta\|_2 \\
= \|M_\sigma^+] \cdot [M_{x_1, \eta_1}; \cdots; M_{x_{|\sigma|}, \eta_{|\sigma|}}]\|_2 \\
= \left\| (M_\sigma^T M_\sigma)^{-1} \sum_{i=1}^{|\sigma|} M_{x_i, \eta_i} \right\|_2 \\
\leq \max_{\eta_1, \cdots, \eta_{|\sigma|} \in [-1,1]^d} \left\| (M_\sigma^T M_\sigma)^{-1} \sum_{i=1}^{|\sigma|} M_{x_i, \eta_i} \right\|_2 \\
= \beta_\sigma
\]

\[ \Box \]

B.2 Proof of Theorem 1

In order to prove Theorem 1, we first present the following three lemmas, Lemma 2-4.

**Lemma 2.** For Algorithm 1, after $n$ exploration rounds,

\[
\Pr[\|\theta - \hat{\theta}(n)\|_2 \geq \text{rad}_n] \leq \frac{\delta}{2n^2}
\]
Proof. In Lemma A.3 in [37], let $\gamma = \text{rad}_n$. Then, we have
\[
\Pr[\|\theta - \hat{\theta}(n)\|_2 \geq \text{rad}_n] \leq 2e^2 \exp \left\{ -\frac{n}{2\beta^2} \frac{2\beta^2 \log(\frac{4n^2e^2}{\delta})}{n} \right\} = \frac{\delta}{2n^2}.
\]

Define the following events
\[
\mathcal{E}_n := \{\forall x \in \mathcal{X}, |\bar{r}(x, \theta) - \bar{r}(x, \hat{\theta}(n))| < L_p \cdot \text{rad}_n\}, n \geq 1
\]
\[
\mathcal{E} := \bigcap_{n=1}^{\infty} \mathcal{E}_n.
\]

Lemma 3.
\[
\Pr[\mathcal{E}] \geq 1 - \delta
\]

Proof. From the continuity of the expected reward function,
\[
\bar{r}(x, \theta) - \bar{r}(x, \hat{\theta}(n)) < L_p \cdot \|\theta - \hat{\theta}(n)\|_2.
\]
From Lemma 2, we have that with probability at least $1 - \frac{\delta}{2n^2}$,
\[
\|\theta - \hat{\theta}(n)\|_2 < \text{rad}_n.
\]
Thus, with probability at least $1 - \frac{\delta}{2n^2}$,
\[
\bar{r}(x, \theta) - \bar{r}(x, \hat{\theta}(n)) < L_p \cdot \text{rad}_n.
\]
In other words,
\[
\Pr[\mathcal{E}_n] \geq 1 - \frac{\delta}{2n^2}.
\]
Thus, we have
\[
\Pr[\mathcal{E}] = 1 - \Pr[\bar{\mathcal{E}}]
\]
\[
\geq 1 - \sum_{n=1}^{\infty} \Pr[\bar{\mathcal{E}}_n]
\]
\[
\geq 1 - \sum_{n=1}^{\infty} \frac{\delta}{2n^2}
\]
\[
\geq 1 - \delta.
\]

Lemma 4. Suppose that $\mathcal{E}$ occurs. If $\text{rad}_n < \frac{\Delta_{\min}}{4L_p}$, Algorithm 1 will terminate.

Proof. Suppose that $\mathcal{E}$ occurs. From the definition of $\mathcal{E}$, we have
\[
\bar{r}(\hat{x}, \hat{\theta}(n)) - \bar{r}(\hat{x}^{-}, \hat{\theta}(n)) > \bar{r}(\hat{x}, \theta) - \bar{r}(\hat{x}^{-}, \theta) - 2L_p \cdot \text{rad}_n
\]
\[
= \Delta_{\min} - 2L_p \cdot \text{rad}_n
\]
\[
> 2L_p \cdot \text{rad}_n
\]
Thus, the stop condition holds, and then Algorithm 1 will terminate.
Now we prove Theorem 1.

Proof. First, we prove the correctness of Algorithm 1. From the stop condition, we have that when Algorithm 1 terminates, for all \( x \in \mathcal{X} \setminus \{ \hat{x} \} \),

\[
\bar{r}(\hat{x}, \hat{\theta}(n)) - \bar{r}(x, \hat{\theta}(n)) > 2L_p \cdot \text{rad}_n.
\]

Then, conditioning on \( \mathcal{E} \), when Algorithm 1 terminates, for all \( x \in \mathcal{X} \setminus \{ \hat{x} \} \),

\[
\bar{r}(\hat{x}, \theta) > \bar{r}(\hat{x}, \hat{\theta}(n)) - L_p \cdot \text{rad}_n
\]
\[
> \bar{r}(x, \hat{\theta}(n)) + L_p \cdot \text{rad}_n
\]
\[
> \bar{r}(x, \theta),
\]

which complete the proof of correctness.

Next, we prove the sample complexity of Algorithm 1. Let \( N \) denote the total number of the exploration rounds. If \( N = 1 \), Theorem 1 trivially holds. In the If \( N > 1 \), from Lemma 4, we have that after \( N - 1 \) exploration rounds,

\[
\sqrt{\frac{2\beta^2 \log(4(N-1)^2\varepsilon^2)}{N - 1}} \geq \frac{\Delta_{\text{min}}}{4L_p}
\]

\[
N \leq \frac{32\beta^2 L_p^2}{\Delta_{\text{min}}^2} \log \left( \frac{4N^2\varepsilon^2}{\delta} \right) + 1
\]

Let \( \tilde{H} := \frac{\beta^2 L_p^2}{\Delta_{\text{min}}^2} \). In the following, we prove \( N \leq 655\tilde{H} \log(\frac{\tilde{H}}{\delta}) \). We can write \( N = C\tilde{H} \log(\frac{\tilde{H}}{\delta}) \) for some \( C > 0 \). In order to prove the theorem, it suffices to prove \( C \leq 655 \). Suppose, on the contrary, that \( C > 655 \). Then, we have

\[
N \leq 32\tilde{H} \log \left( \frac{4N^2\varepsilon^2}{\delta} \right) + 1
\]
\[
= 64\tilde{H} \log \left( \frac{2cC\tilde{H} \log(\frac{\tilde{H}}{\delta})}{\delta} \right) + 1
\]
\[
\leq 64\tilde{H} \log(2cC) + 64\tilde{H} \log(\frac{\tilde{H}}{\delta}) + 64\tilde{H} \log \left( \log(\frac{\tilde{H}}{\delta}) \right) + \tilde{H} \log(\frac{\tilde{H}}{\delta})
\]
\[
\leq 64\tilde{H} \log(2cC) + 129\tilde{H} \log(\frac{\tilde{H}}{\delta})
\]
\[
< C\tilde{H} \log(\frac{\tilde{H}}{\delta})
\]
\[
= N,
\]

which makes a contradiction. Thus,

\[
N \leq 655\tilde{H} \log(\frac{\tilde{H}}{\delta}).
\]

Since an exploration round contains \( |\sigma| \leq n \) actions, the total number of samples

\[
T = |\sigma| \cdot N \leq \frac{655\beta^2 L_p^2}{\Delta_{\text{min}}^2} \log(\frac{\beta^2 L_p^2}{\Delta_{\text{min}}^2})
\]

\( \square \)
B.3 Proof of Corollary 1

Proof. In the linear expected reward case, \( \bar{r}(x, \theta) = x^\top \theta \). Then, we can write the continuity of the expected reward function as

\[
\bar{r}(x, \theta) - \bar{r}(x, \hat{\theta}(n)) < \|x\|_2 \cdot \|\theta - \hat{\theta}(n)\|_2.
\]

Since \( \|x\|_2 \leq m, \forall x \in \mathcal{X} \), this theorem follows from Theorem 1 by setting \( L_p = m \).

\( \square \)

B.4 Proof of Lemma 1

Proof. Recall that in Algorithm 6, we choose \( d \) super arms \( \mathcal{X}_\sigma = \{x_1, \ldots, x_d\} \) from \( \mathcal{X} \), such that \( \text{rank}(X) = d \) where \( X = (x_1, \ldots, x_d) \). Then, for any super arm \( z \in \mathcal{X} \), \( z \) can be written as a linear combination of \( x_1, x_2, \ldots, x_d \), i.e.,

\[
z = Xw,
\]

where \( w \in \mathbb{R}^d \) is the vector of coefficients. Let \( \xi_{\min}(A) \) denote the smallest eigenvalue of matrix \( A \). Then, we have

\[
\sum_{k=1}^{d} |w_k| \leq \sqrt{d \left( \sum_{k=1}^{d} w_k^2 \right)} = \sqrt{d w^\top w} \leq \sqrt{d w^\top X^\top X w} \cdot \max_{w' \in \mathbb{R}^d} \sqrt{\frac{w'^\top w'}{w'^\top X^\top X w'}} \\
\leq \sqrt{d w^\top X^\top X w} \cdot \sqrt{\frac{1}{\xi_{\min}(X^\top X)}} = \sqrt{\frac{d}{\xi_{\min}(X^\top X)}} |z|_2 \leq \sqrt{\frac{md}{\xi_{\min}(X^\top X)}} = \sqrt{\frac{md}{\xi_{\min}(XX^\top)}} = \alpha.
\]

Note that in Algorithm 6, we compute \( \lambda_{\mathcal{X}_\sigma}^* = \text{argmin}_{\lambda \in \Delta(\mathcal{X}_\sigma)} \max_{x \in \mathcal{X}_\sigma} x^\top M(\lambda)^{-1} x \) by the entropic mirror descent (Algorithm 8 in Appendix D). \( \lambda_{\mathcal{X}_\sigma}^* \) is the solution to Proposition 1 and satisfies \( \max_{x \in \mathcal{X}_\sigma} x^\top M(\lambda_{\mathcal{X}_\sigma}^*)^{-1} x = d \). Thus, \( \max_{x \in \mathcal{X}_\sigma} \|x\|_{M(\lambda_{\mathcal{X}_\sigma}^*)^{-1}} = \sqrt{d} \). Thus, for any \( z \in \mathcal{X} \) we have

\[
\|z\|_{M(\lambda_{\mathcal{X}_\sigma}^*)^{-1}} = \|w_1 x_1 + \cdots + w_d x_d\|_{M(\lambda_{\mathcal{X}_\sigma}^*)^{-1}} \leq |w_1| \cdot \|x_1\|_{M(\lambda_{\mathcal{X}_\sigma}^*)^{-1}} + \cdots + |w_d| \cdot \|x_d\|_{M(\lambda_{\mathcal{X}_\sigma}^*)^{-1}} \leq |w_1| \cdot \sqrt{d} + \cdots + |w_d| \cdot \sqrt{d} = (|w_1| + \cdots + |w_d|) \sqrt{d} \leq \alpha \sqrt{d}.
\]

\( \square \)
B.5 Proof of Theorem 2

B.5.1 Technical lemmas for Theorem 2

Lemma 5. When \( n \geq \ell_0(\ell \ln(\frac{5|X|}{\delta})) \) where \( \ell \leq 3 \), we have

\[
\Pr[|x^\top \theta - x^\top \hat{\theta}| \leq \varepsilon, \forall x \in X] \geq 1 - \delta.
\]

Proof. We introduce the high probability bound for the estimator \( \hat{\theta} \) as follows.

Proposition 2 (Lemma 10 in Tao et al. [43]). Let \( c_0 = \max\{4L^2, 3\} \). Let \( n \geq \ell \ln(\delta/8) \) where \( \ell \geq d \). For any fixed \( x \in X \), with probability at least \( 1 - \delta \), we have

\[
|x^\top (\theta - \hat{\theta})| \leq \sqrt{2\|x\|_2^2 + 2\sqrt{d}\|x\|_2\|x\|_{M(\lambda)} - 1 + (4 + 2\sqrt{d/\ell})\|x\|_2\|x\|_{M(\lambda)}^{-1}} / \ell.
\]

Since \( \|x\|_2 \leq \sqrt{m} \) and \( \|x\|_{M(\lambda)}^{-1} \leq \alpha \sqrt{d} \) from Lemma 1, applying Proposition 2 for every super arm in \( X \) and via a union bound, we have that when \( n \geq c_0\ell \ln(\frac{5|X|}{\delta}) \) where \( \ell \geq d \),

\[
\Pr \left[ |x^\top \theta - x^\top \hat{\theta}| \leq \sqrt{\frac{2m + 2\alpha\sqrt{md} + (4 + 2\sqrt{d/\ell})\alpha^2d}{\ell}, \forall x \in X} \right] \geq 1 - \delta.
\]

Setting \( \ell \) as \( \ell(\varepsilon) := \frac{4\alpha\sqrt{md} + 4\alpha^2d + \alpha^2\varepsilon d}{\varepsilon^2} \), we have that with probability at least \( 1 - \delta \), \( \forall x \in X \),

\[
|x^\top (\theta - \hat{\theta})| \leq \sqrt{\frac{2m + 2\alpha\sqrt{md} + (4 + 2\sqrt{d/\ell})\alpha^2d}{\ell}} \leq \varepsilon \left( 1 - \frac{\alpha^2d}{4\alpha\sqrt{md} + 4\alpha^2d + \alpha^2\varepsilon d} + \frac{4d}{4\alpha\sqrt{md} + 4\alpha^2d + \alpha^2\varepsilon d} \right) \frac{\alpha^2d}{(4\alpha\sqrt{md} + 4\alpha^2d + \alpha^2\varepsilon d)} \leq \varepsilon,
\]

which completes the proof.

Next, we show the sample complexity bound for the epoch \( q = 0 \).

Lemma 6. With probability at least \( 1 - \delta_0 \), the first epoch \( q = 0 \) in Algorithm 5 satisfies the following properties: (i) epoch \( q = 0 \) ends with \( x^* \in S_{q+1} = \{\hat{x}_1, \ldots, \hat{x}_d\} \); and (ii) the sample complexity is bounded by

\[
O \left( \frac{c_0(\sqrt{md} + \alpha^2d)}{\Delta_{d+1}^2} \ln \delta^{-1} \right).
\]

Proof. For rounds \( r = 1, \ldots \) in the first epoch \( q = 0 \), define events \( F_r := \{|x^\top \theta - x^\top \hat{\theta}| \leq \frac{\varepsilon}{2}, \forall x \in X\} \). Applying Lemma 5, we have \( \Pr[F_r] \leq 1 - \delta_r \). Define event \( F := \bigcap_{r=1}^\infty F_r \). By a union bound, we have \( \Pr[F_r] \geq 1 - \sum_{r=1}^\infty \delta_r \geq 1 - \sum_{r=1}^\infty \frac{\delta_r}{2} \geq 1 - \delta_q \). We condition the remaining proof on event \( F \).

(i) First, we show that the first epoch \( q = 0 \) will end with \( x^* \in S_{q+1} = \{\hat{x}_1, \ldots, \hat{x}_d\} \). Let \( x_1, x_2, \ldots \) denote the super arms ranked by \( x^\top \theta \) for \( x \in X \) (i.e., \( x_1^\top \theta \geq x_2^\top \theta \ldots \)), and we use \( x^* \) and \( x_1 \) interchangeably.

For any round \( r \geq 1 \), \( x_1^\top \hat{\theta}_r \geq x_1^\top \theta - \frac{\varepsilon_r}{2} \geq x^*_1^\top \theta - \frac{\varepsilon_r}{2} \geq x^*_1^\top \hat{\theta}_r - \frac{\varepsilon_r}{2} > x_{d+1}^\top \hat{\theta}_r \). Thus, when the first epoch \( q = 0 \) ends, \( x_1 \in S_{q+1} = \{\hat{x}_1, \ldots, \hat{x}_d\} \).
Let \( r^* \) be the smallest round such that \( \varepsilon_{r^*} < \frac{\Delta_{d+1}}{2} \). In round \( r^* \), for \( x_i \) s.t. \( i \geq d+1 \), \( x_i^\top \theta_{r^*} - x_i^\top \hat{\theta}_{r^*} \geq (x_i^\top \theta - \frac{\varepsilon_{r^*}^2}{2}) - (x_i^\top \theta + \frac{\varepsilon_{r^*}^2}{2}) \geq \Delta_d - \varepsilon_{r^*} > \varepsilon_{r^*} \). Then, the first epoch \( q = 0 \) will end.

(ii) Since the first epoch \( q = 0 \) will end in (or before) round \( r^* \), which is the smallest round such that \( \varepsilon_{r^*} < \frac{\Delta_{d+1}}{2} \), then the sample complexity of the first epoch \( q = 0 \) is bounded by
\[
O \left( \frac{c_0(\alpha \sqrt{md} + \alpha^2 d)}{\varepsilon_{r^*}^2} \ln \left( \frac{|X|}{\delta_{r^*}} \right) \right)
\]
\[
\geq O \left( \frac{c_0(\alpha \sqrt{md} + \alpha^2 d)}{\Delta_{d+1}^2} (\ln \delta^{-1} + \ln |X| + \ln \ln \Delta_{d+1}^{-1}) \right).
\]

\[\square\]

**B.5.2 Proof of Theorem 2**

Proof. Define \( q^* = \lfloor \log_2 d \rfloor \). For epoch \( q = 0 \), applying Lemma 6, the sample complexity is bounded by
\[
O \left( \frac{c_0(\alpha \sqrt{md} + \alpha^2 d)}{\Delta_{d+1}^2} (\ln \delta^{-1} + \ln |X| + \ln \ln \Delta_{d+1}^{-1}) \right).
\]

For epoch \( q \geq 1 \), the PolyALBA algorithm directly calls subroutine ALBA. Applying Lemma 17 in [43], we can bound the sample complexity for epoch \( q \geq 1 \) by
\[
\sum_{q=1}^{q^*} O \left( \frac{c_0 |\frac{d}{\Delta^2} - |\frac{d}{\Delta^2} + 1|}{\Delta^2} \ln \delta^{-1} + \ln |X| + \ln \ln \Delta^{-1}_{\frac{d}{\Delta^2} + 1} \right)
\]
\[
= \sum_{q=1}^{q^*} O \left( \frac{c_0 |\frac{d}{\Delta^2} - |\frac{d}{\Delta^2} + 1|}{\Delta^2} \ln \delta^{-1} + \ln |X| + \ln \ln \Delta^{-1}_{\frac{d}{\Delta^2} + 1} \right)
\]
\[
= \sum_{q=1}^{q^*} O \left( \frac{c_0 |\frac{d}{\Delta^2} - |\frac{d}{\Delta^2} + 1|}{\Delta^2} \ln \delta^{-1} + \ln |X| + \ln \ln \Delta^{-1}_{\frac{d}{\Delta^2} + 1} \right)
\]
\[
= \sum_{q=1}^{q^*-1} \left( \frac{c_0}{\Delta^2} \ln \delta^{-1} + \ln |X| + \ln \ln \Delta^{-1} \right)
\]
\[
= O \left( \frac{c_0}{\Delta^2} (\ln \delta^{-1} + \ln |X| + \ln \ln \Delta^{-1}) \right)
\]

Summing the sample complexity for epoch \( q = 0 \) and \( q \geq 1 \), we obtain the theorem. \[\square\]

**C Analysis of a naive reduction and UCB-based algorithm for CPE-BL**

In this section, we briefly explain a naive reduction to the classic CPE-MB for CPE-BL in the Top-k setting, and note that there is no simple reduction for general CPE-BL by showing an undesirable property of a UCB-based algorithm with a regularized least-square estimator.

We first remark that with only \( O(k) \) more samples, the problem of Top-k identification with full-bandit feedback can be solved by classic Top-k algorithms in which base arms are queried. Suppose that an algorithm has a sample complexity of \( C_{\Delta} \) in classic setting, it yields a complexity of \( \tilde{O}(k \cdot C_{\Delta}) \) for the full-bandit setting, where \( \tilde{O} \) omits some log factors. This is due to the fact that
the unbiased estimate can be obtained for the difference between the two base arms by comparing two $k$-base arm queries with one base arm difference. Formally, with any fixed base arm $i_0$, one can get an unbiased estimate for the gap $\theta_j - \theta_{i_0}$ with $O(k)$ times larger variance by querying two super-arms $S \cup \{j\}$ and $S \cup \{i_0\}$ for $S \subseteq [d]$ such that $|S| = k - 1$ and $j, i_0 \notin S$. Therefore, when $C_{\delta, \Delta}$ has dependence of $\sum_{i \in [d]} \Delta_i^{-2}$, we also have a sample complexity which has dependence of $\sum_{i \in [d]} \Delta_i^{-2}$ for Top-$k$ case with full-bandit feedback. However, for more complex cases such as matroid, matroid intersection, and $s$-$t$ path, we cannot use such a reduction due to its combinatorial constraint.

We show that with a simple modification using the regularized least-square estimator, CLUCB algorithm proposed in [14] can work for CPE-BL for general constraints such as top-$k$, matroid, matroid intersection, and $s$-$t$ path, and it is a polynomial-time $(\epsilon, \delta)$-PAC algorithm. However, we prove that this naive adaption can be sub-optimal and the sample complexity depends on $\frac{1}{\Delta_{\min}}$ in the worst case.

C.1 Preliminary

We call an algorithm fully adaptive if it changes the arm-selection strategy based on the past observation at all rounds. For such an adaptive algorithm, we cannot use the ordinary least-square estimator for $\theta \in \mathbb{R}^n$ since the estimator is no longer unbiased. Instead we will use the regularized least-square estimator. If the sequence of super-arm selections $x_t = (x_1, \ldots, x_t)$ is adaptively determined based on the past observations, the regularized least-square estimator is given by

$$\hat{\theta}_t = (A_{x_t}^\epsilon)^{-1} b_{x_t},$$

where $A_{x_t}^\epsilon$ and $b_{x_t}$ is defined by

$$A_{x_t}^\epsilon = \epsilon I + \sum_{i=1}^{t} x_i x_i^\top, \quad \text{and} \quad b_{x_t} = \sum_{i=1}^{t} x_i r_i \in \mathbb{R}^n.$$ 

for regularization parameter $\epsilon > 0$ and the identity matrix $I$. If we set $\epsilon = 0$ and we are allowed to sample a base arm, i.e., unit vector at all rounds, it is easy to see that $A_{x_t}(i, i) = T_i(t)$ for $i \in [d]$ and $A_{x_t}(i, j) = 0$ for $i \neq j$, where $T_i(t)$ is the number of times that base arm $i$ is sampled before round $t + 1$.

Abbasi-Yadkori et al. [1] showed that the high probability bound for the regularized least-squares estimator $\hat{\theta}$. 

---

**Algorithm 7:** Combinatorial lower-upper naive confidence bound (CLUNCB)

**Input:** Accuracy $\epsilon > 0$, confidence level $\delta \in (0, 1)$

**Initialization:** For each $e \in [d]$, pull $x_e \in \mathfrak{X}$ such that $e \in x_e$ once. Initialize $A_{x_t}^\epsilon$ and $b_t$.

while $\theta_t^\top x_t - \hat{\theta}_t^\top x_t^* \leq \epsilon$ is not true do

$t \leftarrow t + 1;
\hat{x}_t^* \leftarrow \text{argmax}_{x \in \mathfrak{X}} \hat{\theta}_t^\top x;
\text{Set } \text{rad}_t(e) = C_t \sqrt{(A_{x_t}^\epsilon)^{-1}(e, e)} \text{ for all } e \in [d];
\text{for } e = 1, \ldots, d \text{ do}

if $e \in \hat{x}_t^*$ then $\tilde{\theta}_t(e) \leftarrow \hat{\theta}_t(e) - \text{rad}_t(e);
else $\tilde{\theta}_t(e) \leftarrow \hat{\theta}_t(e) + \text{rad}_t(e);

$\hat{x}_t \leftarrow \text{argmax}_{x \in \mathfrak{X}} \tilde{\theta}_t^\top x;
\tilde{p}_t \leftarrow \text{argmax}_{e \in (\hat{x}_t \setminus \hat{x}_t^*) \cup (\hat{x}_t^* \setminus \tilde{p}_t)} \text{rad}_t(e);
\text{Sample any } x_t \in \mathfrak{X} \text{ such that } p_t \in x_t;
\text{Update } A_{x_t}^\epsilon, b_t \text{ and } \tilde{\theta}_t;
\text{endif}
\text{endfor}
\text{Return } \text{Out} \leftarrow \hat{x}_t^*$
Proposition 3 (Theorem 2 in Abbasi-Yadkori et al. [1]). Let $\hat{\theta}_t$ be the regularized least-squares estimator. Suppose that a noise $\eta_t$ is $\kappa$-sub-Gaussian. If the $\ell_2$-norm of parameter $\theta$ is less than $L$, then for all $i \in [d]$ and for every adaptive sequence $x_t$,

$$|x^T \theta - x^T \hat{\theta}_t| \leq C_t \|x\|_{(A_{x_t})}^{-1}$$

holds for all $t \in \{1, 2, \ldots\}$ and $\forall x \in \mathbb{R}^n$ with probability at least $1 - \delta$, where

$$C_t = \kappa \sqrt{2 \log \frac{\det(A_{x_t})^{1/2}}{t^{1/2}}} + t^{1/2}L.$$  

Moreover, if $\|x\|_2 \leq \sqrt{m}$ holds for all $t > 0$, then

$$C_t \leq \kappa \sqrt{d \log \frac{1 + tm/\iota}{\delta}} + t^{1/2}L.$$  

We also introduce the notion of the width for a decision set $\mathcal{X}$ defined in Chen et al. [14]: width($\mathcal{X}$) prescribes the size of the thinnest exchange class (see [14] for detailed definition). For example, if $\mathcal{X}$ are independent sets of ground set $[d]$, width($\mathcal{X}$) $\leq 2$.

C.2 Analysis for CLUCB with full-bandit feedback

In the setting where each base arm can be pulled, the confidence radius is simply defined as $rad_i(e) = \sqrt{2 \log \frac{4\delta^2}{T_{e}(t)}}$ for all $e \in [d]$. Since we are not allowed to pull each base arm, we can not define such a radius as the above form. However, we have concentration inequalities for each unit vector of $e$, and thus we can construct the confidence radius in the full-bandit setting. From the Proposition 3, we can construct the high probability confidence radius as follows.

Lemma 7. Suppose that a reward from each base arm follows 1-sub-Gaussian distribution for all $i \in [d]$. And if, for all $t > 0$ and all $i \in [d]$, the confidence radius $rad_i(i)$ is defined as

$$rad_i(i) = C_t \sqrt{(A_{x_t})^{-1}(i,i)} \quad (\forall i \in [d]),$$

where $C_t$ is given by (4). Let $\text{rad}_i$ be an $d$-dimensional vector with nonnegative entries. For $\text{rad}_i$, define random event $\mathcal{E}_t$ for all $t > 0$ as follows.

$$\mathcal{E}_t = \{\forall i \in [d], |\theta(i) - \hat{\theta}_t(i)| \leq \text{rad}_t(i)\}$$

Then we have

$$\Pr \left[ \bigcap_{t=1}^{\infty} \mathcal{E}_t \right] \geq 1 - \delta.$$  

The proof is omitted since it is straightforward from Proposition 3 and union bounds. Using the above confidence radius, we can design CLUCB-based algorithm for CPE-BL, which is detailed in Algorithm 7. We show that Algorithm 7 is $(\epsilon, \delta)$-PAC and its sample complexity bound is given in Corollary 2. As can be seen, the sample complexity depends on $\Delta_{\min}^2$ in the worst case. Also, since CLUNCB is fully adaptive, we cannot completely control $\lambda_C$ beforehand and thus $M(M^{-1}(\lambda_C)/\epsilon, \epsilon) \leq \lambda_{\max}(M(M^{-1}(\lambda_C))^{-1})$ can be large.

Corollary 2. Let $\lambda_C \in \Delta(\mathcal{X})$ be a distribution in which $\lambda_C(x)$ represents the ratio that $x$ was pulled by CLUNCB. The total number of samples $T$ is bounded as

$$T = \max_{e} \tau_e = O \left( dk^2 H \log \left( \frac{dk^2 m H/\epsilon + \log \delta^{-1}}{\delta} \right) \right),$$

20
where \( \hat{H} \) is defined as

\[
\hat{H} = \max_{e \in [d]} \left( \frac{M(\lambda_C)^{-1}(e, e) \min \left\{ \frac{9\text{width}(\mathcal{X})^2}{\Delta^2}, \frac{4m^2}{\epsilon^2} \right\}}{\epsilon} \right) \tag{10}
\]

**Proof.** First, we state the following two lemmas in Chen et al. [14]: Lemma 8 (Lemma 12 in Chen et al. [14]) shows that if the confidence radius are valid, then CLUCB always outputs \( \epsilon \)-optimal set, and Lemma 9 (Lemma 13 in Chen et al. [14]) implies that if the confidence radius of an arm is sufficiently small, then the arm will not be chosen as \( p_t \). Note that we have the lemmas since Lemma 3, 5, 7, and 10 in Chen et al. [14] also hold for our setting where \( \text{rad}_t \) is given by (6) and the empirical mean is replaced with the least square estimator \( \hat{\theta}_t \) in (2).

**Lemma 8** (Lemma 12 in Chen et al. [14]). If CLUCB stops on round \( t \) and suppose that \( \mathcal{E}_t \) occurs. Then, we have \( \theta^1 x^* - \theta^1 x_{\text{Out}} \leq \epsilon \).

**Lemma 9** (Lemma 13 in Chen et al. [14]). Given any \( t \) and suppose that event \( \mathcal{E}_t \) occurs. For any \( e \in [d] \), if \( \text{rad}_t(e) < \max \left\{ \frac{\Delta}{3\text{width}(\mathcal{X})}, \frac{\epsilon}{2m} \right\} \), then \( p_t \neq e \).

The random event \( \bigcap_{t=1}^{\infty} \mathcal{E}_t \) occurs with probability at least \( 1 - \delta \) from Lemma 7. From Lemma 8, under the event \( \bigcap_{t=1}^{\infty} \mathcal{E}_t \), CLUCB returns an \( \epsilon \)-optimal set. Therefore, in the rest of part, we shall assume this event holds.

Fix any arm \( e \in [d] \) and let \( \tau_e \) be the last round which arm \( e \) is chosen as \( p_t \). From (5) and for a small \( \iota \leq \frac{\epsilon^2}{2m} \log \delta^{-1} \), we have

\[
C_{\tau_e} \leq \kappa \sqrt{n \log \frac{1 + \tau_e m / \iota}{\delta}} + \iota^4 L \leq 2 \kappa \sqrt{d \log \left( \frac{1 + \tau_e m / \iota}{\delta} \right)} \tag{11}
\]

By Lemma 9, we have \( \text{rad}_{\tau_e}(e) \geq \max \left\{ \frac{\Delta}{3\text{width}(\mathcal{X})}, \frac{\epsilon}{2m} \right\} \). We define \( \Lambda_{A_{\tau_e}} = \frac{A_{\tau_e}^2}{\epsilon} \). Then, we have

\[
\max \left\{ \frac{\Delta^2}{9\text{width}(\mathcal{X})^2}, \frac{\epsilon^2}{4m^2} \right\} \leq \text{rad}_t^2(e) \tag{12}
\]

\[
= C_{\tau_e}^2 (A_{\tau_e}^2)^{-1}(e, e) \tag{13}
\]

\[
\leq 4\kappa^2 d \log \left( \frac{1 + \tau_e m / \iota}{\delta} \right) (A_{\tau_e}^2)^{-1}(e, e) \tag{14}
\]

\[
= 4\kappa^2 d \log \left( \frac{1 + \tau_e m / \iota}{\delta} \right) \frac{(A_{\tau_e}^2)^{-1}(e, e)}{\tau_e} \tag{15}
\]

That is, we obtain

\[
\tau_e \leq 4\kappa^2 d (\Lambda_{A_{\tau_e}})^{-1}(e, e) \min \left\{ \frac{9\text{width}(\mathcal{X})^2}{\Delta^2}, \frac{4m^2}{\epsilon^2} \right\} \log \left( \frac{1 + \tau_e m / \iota}{\delta} \right) = H_e \log \left( \frac{1 + \tau_e m / \iota}{\delta} \right) \tag{16}
\]

where we define \( H_e = 4\kappa^2 d (\Lambda_{A_{\tau_e}})^{-1}(e, e) \min \left\{ \frac{9\text{width}(\mathcal{X})^2}{\Delta^2}, \frac{4m^2}{\epsilon^2} \right\} \). Let \( \tau'(\leq \tau_e) \) satisfying

\[
\tau_e = H_e \left( \log \left( \frac{1 + \tau' m / \iota}{\iota} \right) + \log \frac{1}{\delta} \right) \tag{17}
\]

Then, we have

\[
\tau' \leq \tau_e = H_e \left( \log \left( \frac{1 + \tau' m / \iota}{\iota} \right) + \log \frac{1}{\delta} \right) \tag{18}
\]

\[
\leq H_e \left( \sqrt{\frac{\tau' m}{\iota}} + \log \frac{1}{\delta} \right) \tag{19}
\]
Algorithm 8: The entropic mirror descent for computing $\lambda^*_X$ [43]

**Input** : $d$-set of base arms $[d]$, a set of super arms $X_\sigma \subseteq \mathcal{X}$, Lipschitz constant $L_f$ of function $\log \det M(\lambda)$ and tolerance $\epsilon$

Choose $d$, such that $\text{rank}(X) = d$ where $X = (b_1, \ldots, b_d)$;

Initialize $t \leftarrow 1$ and $\lambda^{(1)} \leftarrow (1/|X_\sigma|, \ldots, 1/|X_\sigma|)$;

while $|\max_{x \in X_\sigma} x^T M(\lambda^{(t)} \lambda_1^{-1}) - d| \geq \epsilon$ do

1. $a_t \leftarrow \sqrt{\frac{2 \ln |X_\sigma|}{L_f \sqrt{t}}}$;
2. Compute gradient $G^{(t)} \leftarrow \text{Tr}(M(\lambda^{(t)})^{-1}(x, x^T))$;
3. Update $\lambda^{(t+1)} \leftarrow \frac{\lambda^{(t)} \exp(a_t G^{(t)})}{\sum_{x \in X_\sigma} \lambda^{(t)} \exp(a_t G^{(t)})}$;
4. $t \leftarrow t + 1$;

$\lambda^*_X \leftarrow \lambda^{(t)}$;

**Output**: $\lambda^*_X$

Solving (20) for $\sqrt{\tau'}$, we obtain

$$\sqrt{\tau'} \leq \frac{1}{2} \left( H_e \sqrt{\frac{m}{t}} + \sqrt{H_e^2 \frac{m}{t} + 4H_e \log \frac{1}{\delta}} \right) \tag{20}$$

$$\leq 2 \sqrt{H_e^2 \frac{m}{4t} + H_e \log \frac{1}{\delta}}. \tag{21}$$

That is, we see that $\tau' = O \left( H_e^2 \frac{m}{t} + H_e \log \frac{1}{\delta} \right)$, which shows that

$$\log \left( \frac{1 + \tau' m}{t} \right) = O \left( \log \left( \frac{mH_e}{t} + \log \frac{1}{\delta} \right) \right). \tag{22}$$

Combining (22) into (17), we obtain

$$\tau_e = O \left( H_e \log \left( \frac{mH_e/t + \log \delta^{-1}}{\delta} \right) \right). \tag{23}$$

The number of samples used by CLUNCB is $T = \max_{x \in [d]} \tau_e$.

Recall that $\lambda_C \in \Delta(\mathcal{X})$ is a distribution in which $\lambda_C(x)$ represents the ratio that $x$ was pulled by CLUNCB. Suppose that $T$ is sufficiently large such that $\Lambda_{x^T} = \frac{\lambda^*_{x^T}}{\lambda_C} \approx M(\lambda_C)$. Define

$$\tilde{H} = \max_{e \in [d]} \left( (M(\lambda_C)^{-1}(e, e) \min \left( \frac{9 \text{width}(\mathcal{X})^2}{\Delta_e^2}, \frac{4m^2}{c d^2} \right) \right) \tag{24}$$

Then, we have

$$T = \max_{e \in [d]} \tau_e = O \left( dk^2 \tilde{H} \log \left( \frac{dk^2 mH_e/t + \log \delta^{-1}}{\delta} \right) \right). \tag{25}$$

\[\square\]

D Approximation algorithm for computing $\lambda \in \Delta(\mathcal{X})$

In this section, we discuss the approximation algorithm for computing $\lambda \in \Delta(\mathcal{X})$. Algorithm 9 is one alternative of Algorithm 6 in PolyALBA. Let $\Delta(\mathcal{X})_{\text{poly}}$ be the subset of probability distributions
Algorithm 9: Approximation algorithm for G-optimal design by the Ellipsoid method

Input: $d$-set of base arms $[d]$, $n \in \mathbb{Z}_+$, $\tilde{n} \in \mathbb{Z}_+$.

for $i = 1, \ldots, n$ do
    Choose $X_{n,i} \leftarrow$ any $n$-super arms;
    Compute $\lambda_i \leftarrow \lambda^*_i$ by Algorithm 8 for $X_{n,i}$;
    Compute $w_{\lambda_i} \in \mathbb{R}^d_+$ by setting $w_{\lambda_i} = (\sum_{j \in [d]} |M(\lambda_i)_{i,j}|^{-1/2})_{j \in [d]} \in \mathbb{R}^d_+$;
end for

Perform the Ellipsoid method to solve the following LP$_{\text{primal}}$:

$$
\text{LP}_\text{primal}: \underset{\nu}{\text{min}} \quad \nu
\quad \text{s.t.} \quad \nu \geq \sum_{i \in [n]} h_i \sum_{e \in [d]} w_{\lambda_i,e} x_e, \ (\forall x \in \mathcal{X})
\quad h \in \triangle([n]).
$$

$h^* \leftarrow$ optimal solution to LP$_{\text{primal}}$;
$\nu^* \leftarrow$ optimal value of LP$_{\text{primal}}$.
Sample $i^* \in [n]$ from $h^* \in \triangle([n])$;

$$
\alpha \leftarrow \min \left\{ \nu^*, \min_{i \in [n]} \frac{md}{\xi_{\min}(M(\lambda_i))} \right\};
$$

Output: $\lambda_{i^*}$ and $\alpha$.

over $\mathcal{X}$ with polynomial-size support. Our task is to find an approximate solution $\tilde{\lambda} \in \triangle(\mathcal{X})_{\text{poly}}$ to the following minmax optimization:

$$
\min_{\lambda \in \triangle(\mathcal{X})} \max_{x \in \mathcal{X}} \|x\|_{M(\lambda)^{-1}}.
$$

We denote the exact G-optimal design by $\lambda^* = \arg\min_{\lambda \in \triangle(\mathcal{X})} \max_{x \in \mathcal{X}} \|x\|_{M(\lambda)^{-1}}$. The above minmax optimization is computationally intractable in combinatorial settings, while many existing methods in linear bandits involved the brute force to solve G-optimal design problem [18, 42, 43]. To avoid a intractable brute force, we address a relaxation problem for the minmax optimization, i.e., a randomized mixed strategy for the robust combinatorial optimization. However, since the ellipsoidal norm $\|x\|_{M(\lambda)^{-1}}$ has the quadratic form, such a relaxation problem is still hard to compute. To overcome this challenge, we consider a simpler norm instead of the ellipsoidal norm. In higher level, this idea is similar to that of Dani et al. [15]: they use skewed octahedron called ConfidenceBall$_1$ as its confidence region rather than the ellipsoid. The radius of ConfidenceBall$_1$ has been set large enough such that it contains the confidence ellipsoid as an inscribed subset. Whereas they use 1-norm $\|M(\lambda)^{1/2}x\|_1$ to define ConfidenceBall$_1$, however, $\max_{x \in \mathcal{X}} \|M(\lambda)^{1/2}x\|_1$ is still intractable in the combinatorial action space. To avoid the computational hardness, we introduce a linear function $g_\lambda : \{0,1\}^d \rightarrow \mathbb{R}_+$ in order to utilize the underlying combinatorial structure. We define $w_\lambda = (\sum_{j \in [d]} |M(\lambda)_{i,j}|^{-1/2})_{i \in [d]} \in \mathbb{R}^d_+$ for $\lambda \in \triangle(\mathcal{X})$. For $\lambda \in \triangle(\mathcal{X})$, a linear function $g_\lambda : \{0,1\}^d \rightarrow \mathbb{R}_+$ is represented as $g_\lambda(x) = \sum_{e \in [d]} w_{\lambda,e} x_e$. We shall assume that the Ellipsoid method computes the optimal solution for linear programmes by enough iterations [22].

We show that the optimal value $\nu^*$ in Algorithm 9 gives an upper bound of the confidence ellipsoidal norm.

Lemma 10. Let $\mathcal{X}$ be a family of super arms satisfying given constraints such as the matroid, matroid intersection, and s-t path. Let $\lambda^{b*} \in \triangle(\mathcal{X})_{\text{poly}}$ be an output by Algorithm 9. Let $h^*$ be an optimal solution and $\nu^*$ be the optimal value for $\text{LP}_{\text{primal}}$. Then, $\lambda^{b*}$ satisfies

$$
\max_{x \in \mathcal{X}} \mathbb{E}[\|x\|_{M(\lambda^{b*})^{-1}}] \leq \nu^*.
$$
Proof. By the definition of $w_\lambda = (\sum_{j \in [d]} |M(\lambda)^{i,j}|^{-1/2})_{i \in [d]}$ and definitions of the quadratic norm and 1-norm, we have

\[
||x||_{M(\lambda)^{-1}} = ||M(\lambda)^{-1/2}x||_2 \\
\leq ||M(\lambda)^{-1/2}x||_1 \\
\leq \sum_{e \in [d]} w_{\lambda,e}x_e \\
= g_\lambda(x) \ (\forall x \in \{0,1\}^d).
\]

Let $y^* = \arg\max_{x \in \mathcal{X}} ||x||_{M(\lambda)^{-1}}$. From Eq. (29), it holds that

\[
\max_{x \in \mathcal{X}} ||x||_{M(\lambda)^{-1}} = ||y^*||_{M(\lambda)^{-1}} \leq g_\lambda(y^*).
\]

Recall that $\lambda^{h^*} = \lambda_i^*$, where $i^*$ was sampled from $h^* \in \triangle([n])$ in Algorithm 9; we have $E[w_{\lambda^{h^*},e}] = \sum_{i \in [n]} h_i^* w_{\lambda_i,e}$ for all $e \in [d]$. Thus, for any $x \in \mathcal{X}$, we see that

\[
E[g_{\lambda^{h^*}}(x)] = \sum_{i \in [n]} h_i^* g_{\lambda_i}(x).
\]

From the above, we have that

\[
\max_{x \in \mathcal{X}} E[||x||_{M(\lambda^{h^*})^{-1}}] \leq \max_{x \in \mathcal{X}} E[g_{\lambda^{h^*}}(x)] = \max_{x \in \mathcal{X}} \sum_{i \in [n]} h_i^* g_{\lambda_i}(x).
\]

Thus, we obtain

\[
\max_{x \in \mathcal{X}} E[||x||_{M(\lambda^{h^*})^{-1}}] \leq \max_{x \in \mathcal{X}} \sum_{i \in [n]} h_i^* g_{\lambda_i}(x) \\
= \min\left\{h \in \triangle([n]) : x \in \mathcal{X}\right\} \sum_{i \in [n]} h_i g_{\lambda_i}(x) \\
= \nu^*
\]

where the first inequality follows by Eq. (31) and the equations follow from the fact that $h^*$ is an optimal solution for LP$_{\text{primal}}$. \hfill \square

Using the above property, $\alpha$ can be replaced with $\alpha = \min \left\{\frac{\nu^*}{\sqrt{d}}, \min_{i \in [n]} \sqrt{\frac{md}{\xi_{\min}(M(\lambda_i))}}\right\}$ in the expected sample complexity given in Theorem 2.

Corollary 3. Let $\nu^*$ be the optimal value for LP$_{\text{primal}}$ obtained in Algorithm 9. With probability at least $1 - \delta$, the PolyALBA algorithm (Algorithm 5) will return the best super arm $x^*$ with an expected sample complexity

\[
O\left(\sum_{i=2}^{d} \frac{c_0}{\Delta_i^2} (\ln \delta^{-1} + \ln |\mathcal{X}| + \ln \Delta_i^{-1}) + \frac{c_0(\alpha \sqrt{md} + \alpha^2 d)}{\Delta_{d+1}^2} (\ln \delta^{-1} + \ln |\mathcal{X}| + \ln \Delta_{d+1}^{-1})\right),
\]

where $\alpha = \min \left\{\frac{\nu^*}{\sqrt{d}}, \min_{i \in [n]} \sqrt{\frac{md}{\xi_{\min}(M(\lambda_i))}}\right\}$.

Note that Algorithm 9 runs in polynomial time as long as $g_\lambda(x)$ is linear and not necessarily $g_\lambda(x) = \sum_{e \in [d]} w_{\lambda,e} x_e$ defined in this section.
**LP-based algorithm for combinatorial robust optimization.** We briefly explain polynomial-time solvability of LP\textsubscript{primal} in Algorithm 9 by the Ellipsoid method. Given a family \( \mathcal{X} \) satisfying a combinatorial constraint and \( n \)-set function \( g_{\lambda_1}, \ldots, g_{\lambda_n} : 2^d \to \mathbb{R}_+ \), we describe how to solve the following combinatorial robust optimization:

\[
\min_{h \in \Delta([n])} \max_{x \in X} \sum_{i \in [n]} h_ig_{\lambda_i}(x). \tag{35}
\]

By von Neumann’s minimax theorem, it holds that

\[
\min_{h \in \Delta([n])} \max_{x \in X} \sum_{i \in [n]} h_ig_{\lambda_i}(x) = \max_{p \in \Delta(X)} \min_{i \in [n]} \sum_{x \in X} p_xg_{\lambda_i}(x). \tag{36}
\]

A polytope of \( \mathcal{X} \) is defined as \( P(\mathcal{X}) = \text{conv}\{x : x \in \mathcal{X}\} \). For a vector \( x \in \mathcal{X} \) we define \( S^x \) to be the corresponding subset form, i.e. \( S^x = \{i \in [n] : x_i = 1\} \). The key observation is that for any distribution \( p \in \Delta(\mathcal{X}) \), we can obtain a point \( y \in P(\mathcal{X}) \) by \( y = p^\top x \), and thus for every \( e \in [d] \), \( y_e = \sum_{x \in \mathcal{X} : e \in S^x} p_x \). This means that \( y_e \) is the marginal probability of seeing an included dimension (a.k.a. a base arm \( e \)) when selecting vector \( x \) (a.k.a. super arm \( S^x \)) according to distribution \( p \).

Then, the optimal value of the above problems is equal to the value of the following LP:

\[
\text{LP}_{\text{dual}}: \text{max. } s \tag{37}
\]

\[
\text{s.t. } s \leq \sum_{e \in [d]} w_{\lambda_e}y_e, \quad (\forall i \in [n]) \tag{38}
\]

\[
y \in P(\mathcal{X}). \tag{39}
\]

If \( \mathcal{X} \) is a matroid, matroid intersection, or the set of \( s-t \) paths, there exists an efficient separation oracle for \( P(\mathcal{X}) \). The separation problem for these constraints can be solved in polynomial time as long as \( g_{\lambda_1}, \ldots, g_{\lambda_n} \) are linear functions. Therefore, due to the theorem of Grötschel et al. \([22]\), we can solve the LP in polynomial time in \( d \) and \( n \). Note that the Ellipsoid method can find an optimal solution to the dual problem of LP\textsubscript{dual}, i.e., LP\textsubscript{primal} in (26). Therefore, we can obtain \( h^* = \arg\min_{h \in \Delta([n])} \max_{x \in X} \sum_{i \in [n]} h_ig_{\lambda_i}(x) \). For the knapsack constraint and the \( r(>2) \)-matroid intersection constraint, the corresponding separation problems are NP-hard. Kawase and Sumita \([31]\) proposed approximation schemes by solving a separation problem for a relaxation of the polytope, which gives PTAS for the knapsack constraint and \( 2/(er) \)-approximate solution for \( r \)-matroid intersection constraint.