Combinatorial Pure Exploration with Full-Bandit or Partial Linear Feedback

Yihan Du,[1*] **Yuko Kuroki,**[2*] **Wei Chen**[3]
[1]IIIS, Tsinghua University, [2]The University of Tokyo, RIKEN, [3]Microsoft Research
duyh18@mails.tsinghua.edu.cn, ykuroki@ms.k.u-tokyo.ac.jp, weic@microsoft.com

Abstract

In this paper, we first study the problem of combinatorial pure exploration with full-bandit feedback (CPE-BL), where a learner is given a combinatorial action space $\mathcal{X} \subseteq \{0,1\}^d$, and in each round the learner pulls an action $x \in \mathcal{X}$ and receives a random reward with expectation $x^\top \theta$, with $\theta \in \mathbb{R}^d$ a latent and unknown environment vector. The objective is to identify the optimal action with the highest expected reward, using as few samples as possible. For CPE-BL, we design the first *polynomial-time adaptive* algorithm, whose sample complexity matches the lower bound (within a logarithmic factor) for a family of instances and has a light dependence of $\Delta_{\min}$ (the smallest gap between the optimal action and sub-optimal actions). Furthermore, we propose a novel generalization of CPE-BL with flexible feedback structures, called combinatorial pure exploration with partial linear feedback (CPE-PL), which encompasses several families of sub-problems including full-bandit feedback, semi-bandit feedback, partial feedback and nonlinear reward functions. In CPE-PL, each pull of action x reports a random feedback vector with expectation of $M_x\theta$, where $M_x \in \mathbb{R}^{m_x \times d}$ is a transformation matrix for x, and gains a random (possibly nonlinear) reward related to x. For CPE-PL, we develop the first *polynomial-time* algorithm, which simultaneously addresses limited feedback, general reward function and combinatorial action space (e.g., matroids, matchings and s-t paths), and provide its sample complexity analysis. Our empirical evaluation demonstrates that our algorithms run orders of magnitude faster than the existing ones, and our CPE-BL algorithm is robust across different $\Delta_{\min}$ settings while our CPE-PL algorithm is the first one returning correct answers for nonlinear reward functions.

1 Introduction

The problem of *best arm identification* (BAI) is the pure-exploration framework in stochastic multi-armed bandits. In BAI, at each step a learner chooses an arm and observes its reward sampled from an unknown distribution, with the goal of returning the best arm with the highest expected reward using as few exploration steps as possible. This problem abstracts a decision making model in the face of uncertainty with a wide range of applications, and has received much attentions in the literature (Even-Dar, Mannor, and Mansour 2006; Audibert, Bubeck, and Munos 2010; Chen and Li 2015; Kaufmann, Cappé, and Garivier 2016).

In many application domains, possible actions have a certain combinatorial structure. For example, each action may be a size-k subset of keywords in online advertisements (Rusmevichientong and Williamson 2006), or an assignment between workers and tasks in crowdsourcing (Lin et al. 2014), or a spanning tree in communication networks (Huang, Liu, and Ding 2008). To deal with such a combinaotrial action space, the model of *combinatorial pure exploration of multi-armed bandits* (CPE-MB) was first proposed by Chen et al. (2014). In this model, there are d base arms, each of which is associated with an unknown reward distribution, and a collection of *super arms*, each of which is a subset of base arms. A learner plays a base arm at each step and observes its random reward, with the goal of identifying the best super arm that maximizes the sum of expected rewards at the end of exploration. CPE-MB generalizes the classical BAI problem (Kalyanakrishnan and Stone 2010; Kalyanakrishnan et al. 2012; Bubeck, Wang, and Viswanathan 2013).

However, many real-world scenarios may not fit into CPE-MB. In particular, 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. Only a few studies avoid such an assumption. Kuroki et al. (2020b) 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 the sum of rewards from the involved base arms. They designed an efficient algorithm for CPE-BL, but the algorithm is nonadaptive and its sample complexity heavily depends on the smallest gap between the best and the other super arms (denoted by $\Delta_{\min}$). Rejwan and Mansour (2020) also designed an efficient algorithm with an adaptive Successive-Accept-Reject algorithm, but the algorithm only works for the top-k case of CPE-BL, which we show can be simply reduced to previous CPE-MB (see Appendix D in the full version).

Note that CPE-BL can be regarded as an instance of the *best arm identification in linear bandits* (BAI-LB), which has received increasing attention recently (Soare, Lazaric, and Munos 2014; Tao, Blanco, and Zhou 2018; Fiez et al. 2019). However, none of the existing algorithms for BAI-LB can efficiently solve CPE-BL, because their running times have

*The first two authors have equal contributions.

polynomial dependence on the size of action space, which is exponential in the combinatorial setting.

In this paper, we provide the first algorithm solving CPE-BL that simultaneous achieves the following properties: (a) polynomial-time complexity, (b) adaptive sampling, such that the sample complexity is not heavily dependent on $\Delta_{\min}$; (c) general combinatorial constraints, and (d) nearly optimal sample complexity for some family of instances.

Next, we propose a more general setting, *combinatorial pure exploration with partial 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 $\mathcal{X} \subseteq \{0,1\}^d$, where each dimension corresponds to a base arm and each action $x \in \mathcal{X}$ can also be viewed as a super arm that contains those dimensions with coordinate 1. At each step the learner chooses an action (super arm) $x_t \in \mathcal{X}$ 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 θ, which may not be a part of the feedback and thus may not be directly observed. Given a confidence level δ, the objective is to identify the optimal action with the maximum expected reward with probability at least $1-\delta$, using as few samples as possible. CPE-PL framework includes CPE-BL as its important sub-problem. 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. For example, in online ranking (Chaudhuri and Tewari 2017), a company recommends their products to users by presenting rankings of entire items, and wants to find the best ranking with limited feedback on the top-ranked item due to user burden constraints and privacy concerns. In crowdsourcing (Lin et al. 2014), an employer assigns crowdworkers to tasks according to the worker-task performance, and wants to find the best matching with limited feedback on a small subset of the completed tasks, owing to the burden of entire feedback and privacy issues (see Section 4.3 for detailed applications).

We remark that, CPE-PL is a novel and general model that encompasses several families of sub-problems across full-bandit feedback, semi-bandit feedback and nonlinear reward function, and it cannot be translated to CPE-BL or BAI-LB. For example, when the reward function is $(x^\top\theta)/\|x\|_1$ and $M_x = \mathrm{diag}(x)$, CPE-PL reduces to a semi-bandit problem with nonlinear reward function, and no existing CPE-BL or BAI-LB algorithm could solve this problem.

Finally, we empirically compare our algorithms with several state-of-the-art CPE-BL and BAI-LB algorithms. Our result demonstrates that (a) our algorithms run much faster than all others, some of which cannot even finish after days of running; (b) For CPE-BL, our adaptive algorithm is much more robust on different $\Delta_{\min}$ settings than the existing non-adaptive algorithm; and (c) For CPE-PL, our algorithm is the only one that correctly outputs the optimal action for a nonlinear reward function among all the compared algorithms.

To summarize, our contributions include: (a) proposing the first *polynomial-time adaptive* algorithm for CPE-BL with general constraints that achieves near optimal sample complexity for some family of instances; and (b) proposing the general CPE-PL framework and the first *polynomial time* algorithm for CPE-PL with its sample complexity analysis.

Due to the space constraint, full proofs with additional results and discussions are moved to the appendices in the full version (Du, Kuroki, and Chen 2020).

1.1 State-of-the-art Related Work

Here we compare with the most related and state-of-the-art works (see Table 1), and the full discussion and comparison table with notation definitions are included in Appendix A in the full version. For CPE-BL, Kuroki et al. (2020b) propose a polynomial-time but static algorithm ICB, which has a heavy dependence on $\Delta_{\min}$ in the sample complexity and requires a large number of samples for small-$\Delta_{\min}$ instances empirically (see Appendix I in the full version). Rejwan and Mansour (2020) develop a polynomial-time adaptive algorithm CSAR but it only works for the top-k case, which has a naive reduction to previous CPE-MB (see Appendix D in the full version).

For BAI-LB where the action space is often considered small, Tao, Blanco, and Zhou (2018) propose an adaptive algorithm ALBA with a light $\Delta_{\min}$ dependence. Fiez et al. (2019) present the first lower bound and a nearly optimal algorithm RAGE. Recently, Katz-Samuels et al. (2020) also design an improved nearly optimal algorithm Peace, which is built upon the previous RAGE. Degenne et al. (2020) and Jedra and Proutiere (2020) develop asymptotically optimal algorithms, but a fair way to compare their results with other non-asymptotical results is unknown. While the existing BAI-LB algorithms achieve satisfactory sample complexity, none of them can efficiently solve CPE-BL with an exponentially large combinatorial action space. This paper proposes the first polynomial-time adaptive algorithm for CPE-BL, which is nearly optimal for some family of instances, and the first polynomial-time algorithm for CPE-PL.

2 Problem Statements

Combinatorial pure exploration with full-bandit linear feedback (CPE-BL). In CPE-BL, a learner is given d base arms numbered $1, 2, \ldots, d$. We define $\mathcal{X} \subseteq \{0,1\}^d$ as a collection of subsets of base arms, which satisfies a certain combinatorial structure such as size-k subsets, matroids, paths and matchings. A subset of base arms $x \in \mathcal{X}$ is called a super arm (or an action). Let m denote the maximum number of base arms that a super arm in $\mathcal{X}$ contains, i.e. $m = \max_{x\in\mathcal{X}} \|x\|_1$ ($m \le d$). There is an unknown environment vector $\theta \in \mathbb{R}^d$ with $\|\theta\|_2 \le L$. At each time step t, a learner pulls a super arm x_t and receives a random reward (full-bandit feedback) $y_t = x^\top(\theta + \eta_t)$, where η_t is a zero-mean noise vector bounded in $[-1,1]^d$ and it is independent among different time step t. Let $x^* = \mathrm{argmax}_{x\in\mathcal{X}}\, x^\top\theta$ denote the optimal super arm, and we assume that the optimal x^* is unique as previous pure exploration works (Chen et al. 2014; Lin et al. 2014; Fiez et al. 2019) do. Let Δ_i denote the gap of the expected rewards between x^* and the super arm with the i-th largest expected reward.

Algorithm	Sample complexity	Case	Problem Type	Strategy	Time
GCB-PE (ours, Thm. 2)	$O\big(\frac{\lvert\sigma\rvert\beta_\sigma^2 L_p^2}{\Delta_{\min}^2}\log\frac{\beta_\sigma^2 L_p^2}{\Delta_{\min}^2\delta}\big)$	General	CPE-PL	Static	$\mathrm{Poly}(d)$
PolyALBA (ours, Thm. 1)	$\tilde{O}\big(\sum_{i=2}^{\lfloor\frac{d}{2}\rfloor}\frac{1}{\Delta_i^2}\log\frac{\lvert\mathcal{X}\rvert}{\delta}+\frac{d^2 m\xi_{\max}(\widetilde{M}(\lambda)^{-1})}{\Delta_{d+1}^2}\log\frac{\lvert\mathcal{X}\rvert}{\delta}\big)$	General	CPE-BL	Adaptive	$\mathrm{Poly}(d)$
ICB (Kuroki et al. 2020b)	$\tilde{O}\big(\frac{d\xi_{\max}(M(\lambda)^{-1})\rho(\lambda)}{\Delta_{\min}^2}\log\frac{d\xi_{\max}(M(\lambda)^{-1})\rho(\lambda)}{\Delta_{\min}^2\delta}\big)$	General	CPE-BL	Static	$\mathrm{Poly}(d)$
CSAR (Rejwan and Mansour 2020)	$\tilde{O}\big(\sum_{i=2}^{d}\frac{1}{\Delta_i^2}\log\frac{d}{\delta}\big)$	Top-k	CPE-BL	Adaptive	$\mathrm{Poly}(d)$
ALBA (Tao, Blanco, and Zhou 2018)	$\tilde{O}\big(\sum_{i=2}^{d}\frac{1}{\Delta_i^2}(\log\delta^{-1}+\log\lvert\mathcal{X}\rvert)\big)$	$\mathcal{X}\subseteq\mathbb{R}^d$	BAI-LB	Adaptive	$\Omega(\lvert\mathcal{X}\rvert)$
RAGE (Fiez et al. 2019)	$O\big(\sum_{t=1}^{\lfloor\log_2(4/\Delta_{\min})\rfloor}2(2^t)^2\tilde{\rho}(\mathcal{Y}(S_t))\log(t^2\lvert\mathcal{X}\rvert^2/\delta)\big)$	$\mathcal{X}\subseteq\mathbb{R}^d$	BAI-LB	Adaptive	$\Omega(\lvert\mathcal{X}\rvert)$
LinGame(-C) (Degenne et al. 2020)	$\limsup_{\delta\to0}\frac{\mathbb{E}_\theta[\tau_\delta]}{\log(1/\delta)}\le\min_{\lambda\in\triangle(\mathcal{X})}\max_{x\in\mathcal{X}\setminus\{x^*\}}\frac{2\lVert x^*-x\rVert^2_{M(\lambda)^{-1}}}{((x^*-x)^\top\theta)^2}$	$\mathcal{X}\subseteq\mathbb{R}^d$	BAI-LB	Adaptive	$\Omega(\lvert\mathcal{X}\rvert)$
Peace (Katz-Samuels et al. 2020)	$O\big((\min_{\lambda\in\triangle(\mathcal{X})}\max_{x\in\mathcal{X}\setminus\{x^*\}}\frac{\lVert x^*-x\rVert^2_{M(\lambda)^{-1}}}{((x^*-x)^\top\theta)^2}+\gamma^*)\log(1/\delta)\big)$	$\mathcal{X}\subseteq\mathbb{R}^d$	BAI-LB	Adaptive	$\Omega(\lvert\mathcal{X}\rvert)$
LT&S (Jedra and Proutiere 2020)	$\limsup_{\delta\to0}\frac{\mathbb{E}_\theta[\tau_\delta]}{\log(1/\delta)}\le\min_{\lambda\in\triangle(\mathcal{X})}\max_{x\in\mathcal{X}\setminus\{x^*\}}\frac{\lVert x^*-x\rVert^2_{M(\lambda)^{-1}}}{((x^*-x)^\top\theta)^2}$	$\mathcal{X}\subseteq\mathbb{R}^d$	BAI-LB	Adaptive	$\Omega(\lvert\mathcal{X}\rvert)$
Lower Bound (Fiez et al. 2019)	$\mathbb{E}_\theta[\tau_\delta]\ge\min_{\lambda\in\triangle(\mathcal{X})}\max_{x\in\mathcal{X}\setminus\{x^*\}}\frac{\lVert x^*-x\rVert^2_{M(\lambda)^{-1}}}{((x^*-x)^\top\theta)^2}\log(1/2.4\delta)$	$\mathcal{X}\subseteq\mathbb{R}^d$	BAI-LB	-	-

Table 1. Comparison between our results and state-of-the-art results for CPE-BL(PL). "General" represents that the algorithm works for any combinatorial structure. $\tilde{O}(\cdot)$ only omits $\log\log$ factors. Main notations is defined in Section 2.

Given a confidence level $\delta\in(0,1)$, the objective is to use as few samples as possible to identify the optimal super arm with probability at least $1-\delta$. This is often called the fixed confidence setting in the bandit literature, and the number of samples required by the learner is called *sample complexity*.

Combinatorial pure exploration with partial-monitoring linear feedback (CPE-PL). CPE-PL is a generalization of CPE-BL to partial linear feedback and nonlinear reward functions. In CPE-PL, each super arm $x\in\mathcal{X}$ is associated with a transformation matrix $M_x\in\mathbb{R}^{m_x\times d}$, whose row dimension m_x depends on x. At each timestep t, a learner pulls a super arm x_t and observes a random linear feedback vector $y_t=M_{x_t}(\theta+\eta_t)\in\mathbb{R}^{m_{x_t}}$, where η_t is the noise vector. 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. Similarly, given a confidence $\delta\in(0,1)$, the learner aims to use as few samples as possible to identify the optimal super arm with probability at least $1-\delta$.

CPE-PL allows more flexible feedback structures than CPE-BL or BAI-LB, and encompasses several families of sub-problems including full-bandit feedback, semi-bandit feedback and nonlinear reward functions. For example, when $M_x=x^\top\in\mathbb{R}^{1\times d}$ for all $x\in\mathcal{X}$, this model reduces to CPE-BL. When $M_x=\mathrm{diag}(x)\in\mathbb{R}^{d\times d}$ for all $x\in\mathcal{X}$, this model reduces to combinatorial pure exploration with semi-bandit feedback (see Appendix B in the full version for illustration examples).

The regret minimization version of CPE-PL has been studied in Lin et al. (2014); Chaudhuri and Tewari (2016). In this paper, we study the pure exploration version and inherit the two technical assumptions from Lin et al. (2014); Chaudhuri and Tewari (2016) in order to design an efficient algorithm.

Assumption 1 (Lipschitz continuity of the expected reward function)**.** *There exists a constant L_p such that for any $x\in\mathcal{X}$ and any $\theta_1,\theta_2\in\mathbb{R}^d$, $|\bar{r}(x,\theta_1)-\bar{r}(x,\theta_2)|\le L_p||\theta_1-\theta_2||_2$.*

Assumption 2 (Global observer set)**.** *There exists a global observer set $\sigma=\{x_1,x_2,\ldots,x_{|\sigma|}\}\subseteq\mathcal{X}$, such that the stacked $\sum_{i=1}^{|\sigma|}m_{x_i}\times d$ transformation matrix $M_\sigma=(M_{x_1};M_{x_2};\ldots;M_{x_{|\sigma|}})$ is of full column rank ($rank(M_\sigma)=d$).*

Then, the Moore-Penrose pseudoinverse M_σ^+ satisfies $M_\sigma^+M_\sigma=I_d$, where I_d is the $d\times d$ identity matrix. We justify Assumption 2 by the fact that without the existence of global observer set, the learner cannot recover θ and may not distinguish two different actions. With Assumption 2, we can systematically construct a global observer set with $|\sigma|\le d$ by sequentially adding an action that strictly increases the rank of M_σ, until M_σ reaches the full rank. Section 4.3 provides more detailed discussion on the global observer set with applications of CPE-PL.

Notations. For clarity, we also introduce the following notations. Let $[d]=\{1,2,\ldots,d\}$. For a vector $x\in\mathbb{R}^d$ and a matrix $B\in\mathbb{R}^{d\times d}$, let $\|x\|_B=\sqrt{x^\top Bx}$. For a positive definite matrix $B\in\mathbb{R}^{d\times d}$, we use $B^{1/2}$ to denote the unique positive definite matrix whose square is B. For a given family $\mathcal{X}$, we use $\triangle(\mathcal{X})$ to denote the set of probability distributions over $\mathcal{X}$. For distribution $\lambda\in\triangle(\mathcal{X})$, we define $\mathrm{supp}(\lambda)=\{x:\lambda(x)>0\}$, $M(\lambda)=\mathbb{E}_{z\sim\lambda}[zz^\top]$ and $\widetilde{M}(\lambda)=\sum_{x\in\mathrm{supp}(\lambda)}xx^\top$. We denote the maximum (minimal) eigenvalue of matrix B by $\xi_{\max}(B)$ ($\xi_{\min}(B)$).

3 Combinatorial Pure Exploration with Full-bandit Feedback (CPE-BL)

In this section, we propose the first polynomial-time adaptive algorithm PolyALBA for CPE-BL, and show that its sample complexity matches the lower bound (within a logarithmic factor) for a family of instances.

3.1 Algorithm Procedure

ALBA algorithm. Before stating the main algorithm, we introduce the Adaptive Linear Best Arm (ALBA) algorithm for BAI-LB (Tao, Blanco, and Zhou 2018) (see Algorithm 1 for

Algorithm 1: ALBA(S, δ) (Tao, Blanco, and Zhou 2018)

Input : Action set S and confidence δ.

Initialize $S_1 \leftarrow S$;
for $q \leftarrow 1, \ldots, \lfloor \log_2 d \rfloor$ **do**
  $\delta_q \leftarrow \frac{6}{\pi^2} \frac{\delta}{(q+1)^2}$;
  $S_{q+1} \leftarrow \mathrm{ElimTil}_{\lfloor \frac{d}{2^q} \rfloor}(S_q, \delta_q)$;
  $q \leftarrow q + 1$;

Output : $x \in S_{q+1}$

Algorithm 2: ElimTil$_p(S, \delta)$

Input : A parameter p, arms set S and confidence level δ.

Compute $\lambda_S^* \leftarrow \min_{\lambda \in \triangle(S)} \max_{x \in S} x^\top M(\lambda)^{-1} x$;
Initialize $S_1 \leftarrow S$, $r \leftarrow 1$;
while $|S_r| > p$ **do**
  Set $\varepsilon_r \leftarrow 1/2^r$, $\delta_r \leftarrow 6/\pi^2 \cdot \delta / r^2$;
  $\hat{\theta}_r \leftarrow \mathsf{VectorEst}(\lambda_S^*, c_0 \frac{2+(6+\varepsilon_r/2)d}{(\varepsilon/2)^2} \ln \frac{5|S|}{\delta_r})$;
  $x_r \leftarrow \mathrm{argmax}_{x \in S_r} x^\top \hat{\theta}_r$;
  $S_{r+1} \leftarrow S_r \setminus \{x \in S_r \mid x^\top \hat{\theta}_r < x_r^\top \hat{\theta}_r - \varepsilon_r\}$;
  $r \leftarrow r + 1$;

Output : S_r

its description), which is the key subroutine of our proposed method PolyALBA. First, we describe the randomized least-square estimator defined by Tao et al. (2018). Let $y_1, \ldots, y_n$ be n $i.i.d.$ samples following a given distribution $\lambda \in \triangle(\mathcal{X})$, and let the corresponding rewards be $r_1, \ldots, r_n$ respectively. Let $b = \sum_{i=1}^n r_i y_i$. Then, the randomized estimator $\hat{\theta}$ is given by $\hat{\theta} = A^{-1} b$, where $A = nM(\lambda) \in \mathbb{R}^{d \times d}$ (recall that $M(\lambda) = \mathbb{E}_{z \sim \lambda}[z z^\top]$). The procedure for computing the estimate for θ is described in VectorEst (Algorithm 3). ALBA is an elimination-based algorithm, where in round q it identifies the top $d/2^q$ arms and discards the remaining arms by means of ElimTil$_p$ (Algorithm 2). Note that ALBA(S, δ) runs in time polynomial to $|S|$. However, since in CPE-BL, $|\mathcal{X}|$ is exponential to the instance size, it is infeasible to run ALBA with $S = \mathcal{X}$. Our main contribution is the nontrivial construction of a polynomial sized S_1 to run ALBA with.

Main algorithm. Now we present our proposed algorithm PolyALBA (see Algorithm 4 for its description), in which ALBA is invoked with $S = S_1$ with $|S_1| = d$. Set S_1 is constructed by a novel preparation procedure in the first epoch ($q = 0$). In this preparation epoch, we first compute a fixed distribution $\lambda \in \triangle(\mathcal{X})$ that has a polynomial-size support and a key parameter α (line 1). Then, based on λ we apply static estimation to estimate θ, until we see a big enough gap between the empirically best and $(d+1)$-th best actions (lines 4–13). The empirical top-d actions, excluding those that also have big gaps to the best one, form the set S_1 (lines 10–11), which is used to call ALBA to obtain the final result $\hat{x}^*$.

Note that, computing the empirical best $d + 1$ super arms

Algorithm 3: VectorEst(λ, n)

Input : distribution λ and the number of samples n

Let $y_1 \ldots, y_n$ be the n samples acquired from $\mathrm{supp}(\lambda)$ according to the distribution λ;
Pull arms $y_1, \ldots, y_n$;
Observe the rewards $r_1, \ldots, r_n$;
$A \leftarrow n \cdot \sum_{x \in \mathrm{supp}(\lambda)} \lambda(x) x x^\top$;
$b \leftarrow \sum_{i=1}^n r_i y_i$;

Output : The estimate $\hat{\theta} \leftarrow A^{-1} b$

Algorithm 4: PolyALBA

Input : confidence level δ, $c_0 = \max\{4L^2, 3\}$.

Set $q \leftarrow 0$ and $\delta_q \leftarrow \frac{6}{\pi^2} \frac{\delta}{(q+1)^2}$;
Compute a distribution $\lambda \leftarrow \lambda^*_{\mathcal{X}_\sigma}$ and parameter $\alpha \leftarrow \sqrt{md/\xi_{\min}(\widetilde{M}(\lambda^*_{\mathcal{X}_\sigma}))}$ by Algorithm 5;
$r \leftarrow 1$;
while true **do**
  Set $\varepsilon_r \leftarrow \frac{1}{2^r}$ and $\delta_r \leftarrow \frac{6}{\pi^2} \frac{\delta_q}{r^2}$;
  $\ell(\varepsilon) \leftarrow \frac{2m + 2\alpha\sqrt{m}d + 4\alpha^2 d + \alpha\varepsilon d}{\varepsilon^2}$;
  $\hat{\theta}_r \leftarrow \mathsf{VectorEst}(\lambda, c_0 \ell(\frac{\varepsilon_r}{2}) \ln(\frac{5|\mathcal{X}|}{\delta_r}))$;
  Select $d+1$ actions $\hat{x}_1, \ldots, \hat{x}_d, \hat{x}_{d+1}$ with the highest $d+1$ empirical means $x^\top \hat{\theta}_r$ in all $x \in \mathcal{X}$;
  if $\hat{x}_1^\top \hat{\theta}_r - \hat{x}_{d+1}^\top \hat{\theta}_r > \varepsilon_r$ **then**
    $B_1 \leftarrow \{\hat{x}_1, \ldots, \hat{x}_d\}$;
    $S_1 \leftarrow B_1 \setminus \{x \in B_1 \mid \hat{x}_1^\top \hat{\theta}_r - x^\top \hat{\theta}_r > \varepsilon_r\}$;
    break;
  $r \leftarrow r + 1$;
$\hat{x}^* \leftarrow$ output by ALBA(S_1, δ_1)

Output : $\hat{x}^*$

Algorithm 5: Computing a distribution λ

Input : d-base arms

Choose any d super arms $\mathcal{X}_\sigma = \{b_1, \ldots, b_d\}$ from $\mathcal{X}$, such that $\mathrm{rank}(X) = d$ where $X = (b_1, \ldots, b_d)$;
$\lambda^*_{\mathcal{X}_\sigma} \leftarrow \mathrm{argmin}_{\lambda \in \triangle(\mathcal{X}_\sigma)} \max_{x \in \mathcal{X}_\sigma} x^\top M(\lambda)^{-1} x$ by the entropy mirror descent algorithm of (Tao, Blanco, and Zhou 2018) (see Algorithm 8 in Appendix E in the full version) ;
$\alpha \leftarrow \sqrt{md/\xi_{\min}(\widetilde{M}(\lambda^*_{\mathcal{X}_\sigma}))}$;

Output : $\lambda^*_{\mathcal{X}_\sigma}$ and α

can be done in polynomial time by using *Lawler's k-best procedure* (Lawler 1972). This procedure only requires the existence of the efficient maximization oracle, which is satisfied in many combinatorial problems such as maximum matching, shortest paths and minimum spanning tree. Moreover, the computational efficiency of PolyALBA is not merely owing to the Lawler's k-best procedure. In fact, even if previous BAI-LB algorithms apply the same procedure, they

cannot run in polynomial time since they explicitly maintain exponential-sized action set and sample on distributions with exponential supports. These render heavy computation and memory in every round of previous algorithms. In contrast, we avoid the naive enumeration and sampling on the combinatorial space directly, and instead find empirical top-d actions as representatives through a novel polynomial-time computation procedure.

3.2 Theoretical Analysis

Now we provide the sample complexity bound of PolyALBA.

Theorem 1. *With probability at least $1-\delta$, the PolyALBA algorithm (Algorithm 4) returns the best super arm x^* with sample complexity*

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

where $\alpha = \sqrt{md/\xi_{\min}(\widetilde{M}(\lambda^*_{\mathcal{X}_\sigma}))}$.

Analysis of the statistical and computational efficiency. The first term in Theorem 1 is for the remaining epochs required by subroutine ALBA and the second term is for the preparation procedure. As shown in Theorem 1, our sample complexity bound has lighter dependence on $1/\Delta_{\min}^2$, 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 $\lambda^*_{\mathcal{X}_\sigma}$ and parameter α. Notice that even if we employ a uniform distribution on a polynomial-size support $\mathcal{X}_\sigma \subseteq \mathcal{X}$, i.e., $\lambda_{\mathcal{X}_\sigma} = (1/|\mathcal{X}_\sigma|)_{x\in\mathcal{X}_\sigma}$, computing the maximal confidence bound $\max_{x\in\mathcal{X}} \|x\|_{M(\lambda_{\mathcal{X}_\sigma})^{-1}}$ is NP-hard, while many (UCB-based) algorithms in LB ignore this issue and simply use a brute force method. In contrast, PolyALBA utilizes G-optimal design (Pukelsheim 2006) and 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 $M(\lambda^*_{\mathcal{X}_\sigma})^{-1}$.

Lemma 1. *For $\lambda^*_{\mathcal{X}_\sigma}$ and α obtained by Algorithm 5, it holds that $\max_{x\in\mathcal{X}} \|x\|_{M(\lambda^*_{\mathcal{X}_\sigma})^{-1}} \le \alpha\sqrt{d}$, where $\alpha = \sqrt{md/\xi_{\min}(\widetilde{M}(\lambda^*_{\mathcal{X}_\sigma}))}$.*

From the equivalence theorem for optimal experimental designs (Proposition 2 in Appendix G in the full version), it holds that $\min_{\lambda\in\triangle(\mathcal{X})}\max_{x\in\mathcal{X}}\|x\|_{M(\lambda)^{-1}} = \sqrt{d}$. From this fact and Lemma 1, we see that $\lambda^*_{\mathcal{X}_\sigma}$ is α (≥ 1)-approximate solution to $\min_{\lambda\in\triangle(\mathcal{X})}\max_{x\in\mathcal{X}}\|x\|_{M(\lambda)^{-1}}$ where $\mathcal{X}$ can be defined by general combinatorial constraints. Note that α can be easily obtained by computing $\xi_{\min}(\widetilde{M}(\lambda^*_{\mathcal{X}_\sigma}))$ (recall that $\widetilde{M}(\lambda) = \sum_{x\in\mathrm{supp}(\lambda)} xx^\top$). Therefore, by employing $\lambda^*_{\mathcal{X}_\sigma}$ and a prior knowledge of its approximation ratio α, 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. Note that $\alpha = 1$ if we compute $\min_{\lambda\in\triangle(\mathcal{X})}\max_{x\in\mathcal{X}}\|x\|_{M(\lambda)^{-1}}$ exactly. If we approximately solve it, α is independent on the arm-selection ratio but it can depend on the support of λ. For further discussion on improving α, please see Appendix E in the full version.

Discussion on the optimality. Fiez et al. (2019) give a sample complexity lower bound for BAI-LB (see Table 1) and propose a nearly (within a logarithmic factor) optimal algorithm RAGE with sample complexity of $O\left(\sum_{t=1}^{\lfloor \log_2(4/\Delta_{\min})\rfloor} 2(2^t)^2\tilde{\rho}(\mathcal{Y}(S_t))\log(t^2|\mathcal{X}|^2/\delta)\right)$. Note that the existing lower bound (Fiez et al. 2019) and nearly (or asymptotically) optimal algorithms (Fiez et al. 2019; Degenne et al. 2020; Katz-Samuels et al. 2020) do not consider computational efficiency for combinatorially-large $|\mathcal{X}|$, and the lower bound for polynomial-time CPE-BL algorithms is still an open problem.

When compared to the lower bound (Fiez et al. 2019), there exists a family of instances such that $\Delta_{\lfloor d/2^{(t-2)}\rfloor+1} = 4\cdot 2^{-t}$, $t = 2,3,\ldots,\log_2(\frac{4}{\Delta_{\min}})$, in which our PolyALBA achieves $O(\sum_{t=2}^{\lfloor \log_2(4/\Delta_{\min})\rfloor} 2(2^t)^2\tilde{\rho}(\mathcal{Y}(S_t))\log(t^2|\mathcal{X}|^2/\delta) + md\xi_{\max}(\tilde{M}^{-1}(\lambda))\tilde{\rho}(\mathcal{Y}(S_1))\log(|\mathcal{X}|^2/\delta))$ sample complexity (see Appendix C in the full version for more details). When ignoring a logarithmic factor and with sufficiently small $\Delta_{\min}$, the additional term related to $\xi_{\max}(\tilde{M}^{-1}(\lambda))$ is absorbed and the result matches the lower bound, which shows superiority over other heavily $\Delta_{\min}$-dependent algorithms (Soare, Lazaric, and Munos 2014; Karnin 2016; Kuroki et al. 2020b). Note that the term related to $\xi_{\max}(\tilde{M}^{-1}(\lambda))$ can be viewed as the cost for achieving computational efficiency.

To our best knowledge, our PolyALBA is the first polynomial-time adaptive algorithm that works for CPE-BL with general combinatorial structures and achieves nearly optimal sample complexity for a family of problem instances.

4 Combinatorial Pure Exploration with Partial Linear Feedback (CPE-PL)

In this section, we present the first polynomial-time algorithm GCB-PE for CPE-PL with sample complexity analysis, and discuss its further improvements via a non-uniform allocation strategy. We also give practical applications for CPE-PL and explain the corresponding global observer set and sample complexity result in these scenarios.

4.1 Algorithm Procedure

We illustrate GCB-PE in Algorithm 6. GCB-PE estimates the environment vector θ 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 σ *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 $\vec{y}_n = (y_1; y_2; \ldots; y_{|\sigma|})$. The estimate of environment vector θ in this exploration round is $\hat{\theta}_n = M_\sigma^+ \vec{y}_n$, where M_σ^+ is the Moore-Penrose pseudoinverse of M_σ. From Assumption 2, we have $\mathbb{E}[\hat{\theta}_n] = \theta$. 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 θ.

Similar to Lin et al. (2014), we define a constant $\beta_\sigma := \max_{\eta_1,\cdots,\eta_{|\sigma|}\in[-1,1]^d} \|(M_\sigma^\top M_\sigma)^{-1} \sum_{i=1}^{|\sigma|} M_{x_i}^\top M_{x_i} \eta_i\|_2$, which only depends on global observer set σ, and bounds the estimate error of one exploration round, i.e., for any n, $\|\hat{\theta}_n - \theta\|_2 \le \beta_\sigma$, the proof of which is given in Appendix H.1 in the full version. Based on β_σ, we further design a global confidence radius $\text{rad}_n = \sqrt{2\beta_\sigma^2 \log(4n^2e^2/\delta)/n}$ for the estimate $\hat{\theta}(n)$, and show that with high probability, rad_n bounds the estimate error of $\hat{\theta}(n)$.

Compared with GCB in Lin et al. (2014), 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 rad_n is novelly 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 novelly 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 θ is known.

4.2 Theoretical Analysis

We give the sample complexity of GCB-PE below.

Theorem 2. *With probability at least $1-\delta$, the GCB-PE algorithm (Algorithm 6) will return the optimal super arm x^* with sample complexity*

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

where $|\sigma| \le d$.

When the expected reward function is linear, i.e. $\bar{r}(x,\theta) = x^\top\theta$, we have $L_p = \sqrt{m}$, where m ($\le d$) is the maximum number of base arms a super arm contains. In addition, $\beta_\sigma = \text{Poly}(d)$ in several practical applications of CPE-PL (see Section 4.3 for our detailed discussion).

Discussion on the optimality. While the sample complexity of GCB-PE is sometimes worse than the CPE-BL or BAI-LB algorithms (PolyALBA, ALBA and RAGE), it solves a more general class of problems than CPE-BL and BAI-LB. We emphasize that our contribution mainly focuses on proposing the first polynomial-time algorithm GCB-PE that simultaneously addresses combinatorial action apace, partial linear feedback and nonlinear reward function.

Algorithm 6: GCB-PE

Input : Confidence level δ, global observer set σ, constant β_σ, Lipschitz constant L_p

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

Output : $\hat{x}$

On non-uniform or adaptive allocation strategy. GCB-PE can be further improved by employing a *non-uniform* allocation strategy when considering the global observer set σ with multiplicity: we can obtain such an allocation by solving an optimization $\text{argmin}_{\lambda\in\triangle(\sigma)} \beta_\sigma(\lambda)$ and rounding the result, where $\beta_\sigma(\lambda) := \max_{\eta_1,\cdots,\eta_{|\sigma|}\in[-1,1]^d} \|(M_\sigma^\top M_\sigma)^{-1} \sum_{i=1}^{|\sigma|} \lambda_i M_{x_i}^\top M_{x_i} \eta_i\|_2$. Since uniform sampling is not essential in our analysis, the proposed improvement for GCB-PE via non-uniform allocation does not violate Assumption 2 and keeps our theoretical analysis. GCB-PE is a static algorithm, and we leave the study of adaptive strategies for CPE-PL as future work. In Appendix D in the full version, we discuss a fully-adaptive algorithm for CPE-BL (special case of CPE-PL), and show that the result depends on a non-controllable term $M(\lambda)^{-1}$, which indicates that the static control may be required to deal with linear feedback efficiently.

4.3 Applications for GCB-PL

CPE-PL characterizes more flexible feedback structures than CPE-BL (or BAI-LB) and finds many real-world applications. Below we present two practical applications and discuss the global observer set (Assumption 2) and parameter β_σ.

Online ranking. Consider that a company wishes to recommend their products to users by presenting the ranked list of items. Due to user burden constraints and privacy concerns, collecting a large amount of data on the relevance of all items

might be infeasible, and thus the company usually collects the relevance of only the top-ranked item (Chaudhuri and Tewari 2015, 2016, 2017). In this scenario, a learner selects a permutation of d items (each action x is a permutation) at each step, and observes the relevance of the top-ranked item, i.e., M_x contains a single row with 1 in the place of the top-ranked item and 0 everywhere else. The objective is to identify the best permutation as soon as possible. Then, we can construct a global observable set σ to be the set of any d actions which places a distinct item at top. Here M_σ is the $d \times d$ identity matrix and $\beta_\sigma = \sqrt{d}$.

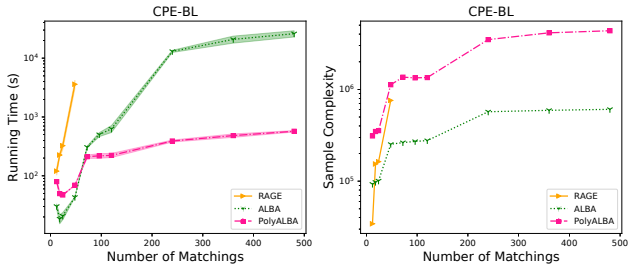
Task assignments in crowdsourcing. Consider that an employer wishes to assign crowdworkers to tasks with high quality performance, and it wants to avoid the high cost and the privacy concern of collecting each individual worker-task pair performance (Lin et al. 2014). Thus, the employer sequentially chooses an assignment from N workers to M tasks (each action x is a worker-task matching) and only collects the sum of performance feedback for $1 \leq s < N$ matched worker-task pairs, i.e., M_x contains a single row with 1s in the places of s matched pairs and 0 everywhere else. The objective is to find the best worker-task matching as soon as possible. For $1 \leq s < N$, Lin et al. (2014) provide a systematic method to construct a global observer set.

5 Experiments

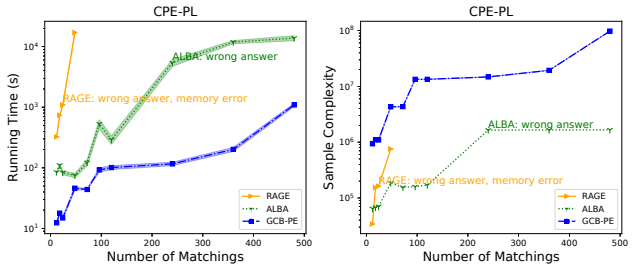
We conduct experiments for CPE-BL and CPE-PL on the matching and top-k instances, and compare our algorithms with the state-of-the-arts in both running time and sample complexity. Due to the space limit, here we only present the results on matchings and defer the top-k results with discussion on $\Delta_{\min}$-dependence to Appendix I in the full version.

We evaluate all the compared algorithms on Intel Xeon E5-2640 v3 CPU at 2.60GHz with 132GB RAM. For both CPE-BL and CPE-PL, we set action space $\mathcal{X}$ as matchings in 3-by-3, 4-by-4 and 5-by-5 complete bipartite graphs. The dimension d, i.e. the number of edges, is set from 9 to 25. The number of matchings $|\mathcal{X}|$ are set from 12 to 480. $\theta_1, \ldots, \theta_d$ is set as a geometric sequence in $[0, 1]$. We simulate the random feedback for action x by a Gaussian distribution with mean of $x^\top \theta$ and unit variance. For CPE-PL, we use the full-bandit feedback as CPE-BL ($M_x = x^\top$) but a nonlinear reward function $\bar{r}(x, \theta) = x^\top \theta / \|x\|_1$. For each algorithm, we perform 20 independent runs and present the average running time and sample complexity with 95% confidence intervals across runs. In the experiments, RAGE (Fiez et al. 2019) reports memory errors when $|\mathcal{X}| > 48$ due to its heavy memory burden, and thus we only obtain its results on small-$|\mathcal{X}|$ instances. For PolyALBA, ALBA and RAGE, we obtain the same sample complexity in different runs, since these algorithms compute the required samples at the beginning of each phase and then perform the fixed samples.

Experiments for CPE-BL. For CPE-BL, we compare our PolyALBA with the state-of-the-art BAI-LB algorithms ALBA and RAGE in running time and sample complexity. As shown in Figure 1(a) with a logarithmic y-axis, our PolyALBA runs about two orders of magnitude faster than



(a) CPE-BL



(b) CPE-PL

Figure 1. Experimental results of running time and sample complexity for CPE-BL and CPE-PL.

ALBA and RAGE, and the running time of PolyALBA increases more slowly than the others as $|\mathcal{X}|$ increases. Due to the extra preparation epoch, PolyALBA has a higher sample complexity, but we argue that in practice one has to keep the computation time low first to make an algorithm useful, and for that matter ALBA and RAGE are too slow to run and PolyALBA is the only feasible option.

Experiments for CPE-PL. For CPE-PL, we compare GCB-PE with BAI-LB algorithms ALBA and RAGE in running time and sample complexity on a more challenging nonlinear reward task. In the experiments for CPE-PL, ALBA and RAGE return wrong answers because they are not designed to handle nonlinear reward functions. Nevertheless, we can still analyze the running times presented in Figure 1(b). It shows that our GCB-PE runs two orders of magnitude faster than ALBA and RAGE while reporting the correct answer. In addition, as $|\mathcal{X}|$ increases, the running time of GCB-PE increases in a much slower pace than the others. The experimental results demonstrate the capability of GCB-PE to simultaneously deal with combinatorial action space, nonlinear reward function and partial feedback in a computationally efficient way.

6 Future Work

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 σ and the distribution $\lambda^*_{\mathcal{X}_\sigma}$, and derive specific sample complexity bounds. Furthermore, the extension of CPE-PL to nonlinear feedback is also a practical and valuable problem.

Acknowledgements

Yuko Kuroki would like to thank Masashi Sugiyama and Junya Honda for helpful comments on the manuscript, and Kento Nozawa for his support to use the server. Yuko Kuroki is supported by Microsoft Research Asia, KAKENHI 18J23034, and JST ACT-X 1124477.

References

Abbasi-Yadkori, Y.; Pál, D.; and Szepesvári, C. 2011. Improved Algorithms for Linear Stochastic Bandits. In *Proc. NIPS'14*, 2312–2320.

Audibert, J.-Y.; Bubeck, S.; and Munos, R. 2010. Best Arm Identification in Multi-Armed Bandits. In *Proc. COLT'10*, 41–53.

Bubeck, S.; Wang, T.; and Viswanathan, N. 2013. Multiple identifications in multi-armed bandits. In *Proc. ICML'13*, 258–265.

Chaudhuri, S.; and Tewari, A. 2015. Online ranking with top-1 feedback. In *Proc. AISTATS'15*, 129–137.

Chaudhuri, S.; and Tewari, A. 2016. Phased Exploration with Greedy Exploitation in Stochastic Combinatorial Partial Monitoring Games. In *Proc. NIPS'16*, 2433–2441.

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

Chen, L.; and Li, J. 2015. On the Optimal Sample Complexity for Best Arm Identification. *arXiv preprint* arXiv:1511.03774.

Chen, S.; Lin, T.; King, I.; Lyu, M. R.; and Chen, W. 2014. Combinatorial Pure Exploration of Multi-Armed Bandits. In *Proc. NIPS'14*, 379–387.

Chen, W.; Du, Y.; Huang, L.; and Zhao, H. 2020. Combinatorial Pure Exploration for Dueling Bandit. In *Proc. ICML'20*, 1531–1541.

Dani, V.; Hayes, T. P.; and Kakade, S. M. 2008. Stochastic Linear Optimization Under Bandit Feedback. In *Proc. COLT'08*, 355–366.

Degenne, R.; Menard, P.; Shang, X.; and Valko, M. 2020. Gamification of Pure Exploration for Linear Bandits. In *Proc. ICML'20*, 2432–2442.

Du, Y.; Kuroki, Y.; and Chen, W. 2020. Combinatorial Pure Exploration with Full-Bandit or Partial Linear Feedback. In *arXiv preprint arXiv:2006.07905*. URL https://arxiv.org/pdf/2006.07905.pdf.

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

Fiez, T.; Jain, L.; Jamieson, K. G.; and Ratliff, L. 2019. Sequential Experimental Design for Transductive Linear Bandits. In *Proc. NeurIPS' 19*, 10667–10677.

Gabillon, V.; Ghavamzadeh, M.; and Lazaric, A. 2012. Best Arm Identification: A Unified Approach to Fixed Budget and Fixed Confidence. In *Proc. NIPS'12*, 3212–3220.

Gabillon, V.; Ghavamzadeh, M.; Lazaric, A.; and Bubeck, S. 2011. Multi-Bandit Best Arm Identification. In *Proc. NIPS'11*, 2222–2230.

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

Huang, S.; Liu, X.; and Ding, Z. 2008. Opportunistic Spectrum Access in Cognitive Radio Networks. In *Proc. INFOCOM'08*, 1427–1435.

Huang, W.; Ok, J.; Li, L.; and Chen, W. 2018. Combinatorial Pure Exploration with Continuous and Separable Reward Functions and Its Applications. In *Proc. IJCAI '18*, 2291–2297.

Jedra, Y.; and Proutiere, A. 2020. Optimal Best-arm Identification in Linear Bandits. *to appear in NeurIPS'20* .

Kalyanakrishnan, S.; and Stone, P. 2010. Efficient Selection of Multiple Bandit Arms: Theory and Practice. In *Proc. ICML'10*, 511–518.

Kalyanakrishnan, S.; Tewari, A.; Auer, P.; and Stone, P. 2012. PAC Subset Selection in Stochastic Multi-armed Bandits. In *Proc. ICML'12*, 655–662.

Karnin, Z. S. 2016. Verification Based Solution for Structured MAB Problems. In *Proc. NIPS' 16*, 145–153.

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. *arXiv preprint arXiv:2006.11685* .

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

Kawase, Y.; and Sumita, H. 2019. Randomized strategies for robust combinatorial optimization. In *Proc. AAAI '19*, 7876–7883.

Kiefer, J.; and Wolfowitz, J. 1960. The Equivalence of Two Extremum Problems. *Canadian Journal of Mathematics* 12: 363–366.

Kuroki, Y.; Miyauchi, A.; Honda, J.; and Sugiyama, M. 2020a. Online Dense Subgraph Discovery via Blurred-Graph Feedback. In *Proc. ICML'20*, 5522–5532.

Kuroki, Y.; Xu, L.; Miyauchi, A.; Honda, J.; and Sugiyama, M. 2020b. Polynomial-Time Algorithms for Multiple-Arm Identification with Full-Bandit Feedback. *Neural Computation* 32(9): 1733–1773.

Lawler, E. L. 1972. 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.

Lin, T.; Abrahao, B.; Kleinberg, R.; Lui, J.; and Chen, W. 2014. Combinatorial partial monitoring game with linear feedback and its applications. In *Proc. ICML'14*, 901–909.

Pukelsheim, F. 2006. *Optimal Design of Experiments*. Society for Industrial and Applied Mathematics.

Rejwan, I.; and Mansour, Y. 2020. Top-k Combinatorial Bandits with Full-Bandit Feedback. In *Proc. ALT' 20*, 752–776.

Rusmevichientong, P.; and Williamson, D. P. 2006. An Adaptive Algorithm for Selecting Profitable Keywords for Search-based Advertising Services. In *Proc. EC '06*, 260–269.

Soare, M.; Lazaric, A.; and Munos, R. 2014. Best-Arm Identification in Linear Bandits. In *Proc. NIPS'14*, 828–836.

Tao, C.; Blanco, S.; and Zhou, Y. 2018. Best Arm Identification in Linear Bandits with Linear Dimension Dependency. In *Proc. ICML'18*, 4877–4886.

Xu, L.; Honda, J.; and Sugiyama, M. 2018. A fully adaptive algorithm for pure exploration in linear bandits. In *Proc. AISTATS'18*, 843–851.

Zaki, M.; Mohan, A.; and Gopalan, A. 2019. Towards Optimal and Efficient Best Arm Identification in Linear Bandits. *arXiv preprint arXiv:1911.01695* .

Zaki, M.; Mohan, A.; and Gopalan, A. 2020. Explicit Best Arm Identification in Linear Bandits Using No-Regret Learners. *arXiv preprint arXiv:2006.07562* .

Zhong, Z.; Cheung, W. C.; and Tan, V. 2020. Best Arm Identification for Cascading Bandits in the Fixed Confidence Setting. In *Proc. ICML'20*, 11481–11491.