Combinatorial Pure Exploration with Continuous and Separable Reward Functions and Its Applications*

Weiran Huang[1,†], **Jungseul Ok**[2], **Liang Li**[3] and **Wei Chen**[4,†]
[1] Tsinghua University, Beijing, China
[2] KTH, Stockholm, Sweden
[3] Ant Financial Group, Hangzhou, China
[4] Microsoft Research, Beijing, China
huang.inbox@outlook.com, jungseul@kth.se, liangli.ll@antfin.com, weic@microsoft.com

Abstract

We study the Combinatorial Pure Exploration problem with Continuous and Separable reward functions (CPE-CS) in the stochastic multi-armed bandit setting. In a CPE-CS instance, we are given several stochastic arms with unknown distributions, as well as a collection of possible decisions. Each decision has a reward according to the distributions of arms. The goal is to identify the decision with the maximum reward, using as few arm samples as possible. The problem generalizes the combinatorial pure exploration problem with linear rewards, which has attracted significant attention in recent years. In this paper, we propose an adaptive learning algorithm for the CPE-CS problem, and analyze its sample complexity. In particular, we introduce a new hardness measure called the consistent optimality hardness, and give both the upper and lower bounds of sample complexity. Moreover, we give examples to demonstrate that our solution has the capacity to deal with non-linear reward functions.

1 Introduction

The stochastic multi-armed bandit model is a predominant model for characterizing the trade-off between exploration and exploitation in a variety of application fields with stochastic environments. In this model, we are given a set of stochastic arms associated with unknown distributions. Upon each play of an arm, the player can get a reward sampled from the corresponding distribution. The most well studied objective is to maximize the cumulative reward, or minimize the cumulative regret, e.g., [Lai and Robbins, 1985; Auer *et al.*, 2002b; Auer *et al.*, 2002a; Bubeck *et al.*, 2012]. Another popular objective is to identify the optimal arm with high probability by adaptively sampling arms based on the feedback collected. This is called the pure exploration version of the multi-armed bandit problem [Bubeck *et al.*, 2011; Audibert and Bubeck, 2010; Gabillon *et al.*, 2012].

Instead of identifying the single optimal arm, there are a class of extended problems identifying the optimal combinatorial decision, e.g., top-k arm identification [Kalyanakrishnan and Stone, 2010; Kalyanakrishnan *et al.*, 2012; Bubeck *et al.*, 2013; Kaufmann and Kalyanakrishnan, 2013; Zhou *et al.*, 2014], multi-bandit best arm identification [Gabillon *et al.*, 2011], and their extension, Combinatorial Pure Exploration with Linear reward functions (CPE-L) [Chen *et al.*, 2014; Chen *et al.*, 2016a], etc. In CPE-L [Chen *et al.*, 2014], the rewards are linear functions on the means of underlying arms, and the decision class is subsets of arms satisfying certain combinatorial constraints.

In this paper, we further generalize CPE-L problems to a large class of Combinatorial Pure Exploration with Continuous and Separable reward functions (CPE-CS) (see Section 2 for the technical definition). We propose the Consistently Optimal Confidence Interval (COCI) algorithm to solve the CPE-CS problem. To analyze its sample complexity, we define a new arm-level measure called *consistent optimality radius* Λ_i of arm i and a new hardness measure called *consistent optimality hardness* $\mathbf{H}_{\mathbf{\Lambda}} = \sum_{i=1}^{m} 1/\Lambda_i^2$, where m is the number of arms. We prove that with probability at least $1-\delta$, COCI finds the optimal solution in $O(\mathbf{H}_{\mathbf{\Lambda}} \log(\mathbf{H}_{\mathbf{\Lambda}} \delta^{-1}))$ rounds. We also show that CPE-CS problems have a lower bound $\Omega(\mathbf{H}_{\mathbf{\Lambda}} + \mathbf{H}_{\mathbf{\Lambda}} m^{-1} \log \delta^{-1})$ in expectation, indicating that the hardness $\mathbf{H}_{\mathbf{\Lambda}}$ is necessary.

We demonstrate the usefulness of CPE-CS by two applications. The first one is water resource planning [Bradley *et al.*, 1977]. The goal is to remove waste at water sources of an area. One can first do some purification tests at different sources to estimate the water quality responses, and then determines the final allocation of purification powers among different sources. One need to balance the trade-off between the purification power and the cost, and usually the objective function is non-linear. This application can be generalized to other urban planning scenarios such as air pollution control, crime control, etc. The second application is partitioned opinion sampling [Bethel, 1986; Ballin and Barcaroli, 2013; Huang *et al.*, 2017]. The opinion polling is done by partitioning people into groups and sampling each group separately with different sample budget to improve the sample quality. One can first do some tests in each group to estimate its opinion variance, and then determines the sample size for each group under the total sample budget for the formal sampling

†Corresponding authors.
*Due to space constraints, supplementary materials and proofs are moved into the extended version [Huang *et al.*, 2018].

process. In this case, the objective function is also non-linear. Furthermore, we show that the COCI algorithm also solves the CPE-L problem with the same sample complexity as the CLUCB algorithm proposed by Chen *et al.* [2014].

In summary, our contributions include: (a) studying the combinatorial pure exploration problem with continuous and separable functions and proposing the COCI algorithm as its solution, (b) analyzing the sample complexity of COCI and providing both its lower and upper bounds with a novel hardness measure, and (c) applying the CPE-CS framework to water resources planning and partitioned opinion sampling with non-linear reward functions to demonstrate the usefulness of the CPE-CS framework and the COCI algorithm.

Related Work. Pure exploration bandit studies adaptive learning methods to identify the optimal solution. Best arm identification [Bubeck *et al.*, 2011; Audibert and Bubeck, 2010; Gabillon *et al.*, 2012], top-k arm identification [Kalyanakrishnan and Stone, 2010; Kalyanakrishnan *et al.*, 2012; Bubeck *et al.*, 2013; Kaufmann and Kalyanakrishnan, 2013; Zhou *et al.*, 2014], the multi-bandit best arm identification [Gabillon *et al.*, 2011] have been studied in the literature. Chen *et al.*; Chen *et al.* [2014; 2016a] generalize these studies to Combinatorial Pure Exploration with Linear reward functions (CPE-L). Soare *et al.* [2014] also study the linear reward functions, but the player is required to select a decision to play instead of a single arm to sample in each round. A very recent paper [Chen *et al.*, 2017] studies the CPE problems beyond linear reward functions, but their model assumes arms with Gaussian distributions and only works with the mean estimator, while our CPE-CS only requires bounded distributions and also works for variance estimators. Moreover, for efficient implementations, they need a pseudo-polynomial algorithm for the exact query besides the maximization oracle, but our solution only needs a maximization oracle.

A related online learning problem is multi-armed bandit (MAB), e.g., [Lai and Robbins, 1985; Auer *et al.*, 2002b; Auer *et al.*, 2002a; Bubeck *et al.*, 2012]. The goal of MAB is to maximize cumulative rewards over multiple rounds, and the key is to balance exploration and exploitation during the learning process. In contrast, in pure exploration, the key is the adaptive exploration in the learning process to quickly find the optimal solution, and thus it is fundamentally different from MAB [Bubeck *et al.*, 2011]. Combinatorial MAB is a popular topic in recent years [Cesa-Bianchi and Lugosi, 2012; Gai *et al.*, 2012; Chen *et al.*, 2016c; Chen *et al.*, 2016b; Gopalan *et al.*, 2014; Kveton *et al.*, 2014; Kveton *et al.*, 2015; Combes *et al.*, 2015], but their goals and techniques are very different from ours.

2 Problem Definition

An instance of combinatorial pure exploration bandit problems consists of (a) a set of m arms $[m] = \{1, \ldots, m\}$, each arm i being associated with an unknown distribution D_i with range $[0, 1]$ and a key unknown parameter $\theta_i^* \in [0, 1]$ of D_i, (b) a finite set of decisions $\mathcal{Y} \subseteq \mathbb{R}^m$, with each decision $\boldsymbol{y} = (y_1, \ldots, y_m)$ as a vector, and (c) a real-valued (expected) reward function $r(\boldsymbol{\theta}; \boldsymbol{y})$ with vector $\boldsymbol{\theta}$ taken from the parameter space $[0, 1]^m$ and $\boldsymbol{y} \in \mathcal{Y}$. In each round $t = 1, 2, \ldots$, a player selects one arm $i \in [m]$ to play, and observes a sample independently drawn from D_i as the feedback. The player needs to decide based on the observed feedback so far if she wants to continue to play arms. If so, she needs to decide which arm to play next; if not, she needs to output a decision $\boldsymbol{y}^o \in \mathcal{Y}$ such that with high probability $\boldsymbol{y}^o$ is the optimal decision maximizing the reward $r(\boldsymbol{\theta}^*; \boldsymbol{y}^o)$, where $\boldsymbol{\theta}^* = (\theta_1^*, \ldots, \theta_m^*)$ is the vector of the true underlying parameters of the unknown distributions $\boldsymbol{D} = (D_1, \ldots, D_m)$.

Definition 1. *Given a combinatorial pure exploration instance* $([m], \mathcal{Y}, r(\cdot;\cdot), \boldsymbol{D}, \boldsymbol{\theta}^*)$ *and a confidence error bound* δ*, the* combinatorial pure exploration (CPE) problem *requires the design of an algorithm with the following components: (a) a stopping condition, which decides whether the algorithm should stop in the current round, (b) an arm selection component, which selects the arm to play in the current round when the stopping condition is false, and (c) an output component, which outputs the decision* $\boldsymbol{y}^o$ *when the stopping condition is true. The algorithm could only use* $([m], \mathcal{Y}, r(\cdot;\cdot))$ *and the feedback from previous rounds as inputs, and should guarantee that with probability at least* $1 - \delta$ *the output* $\boldsymbol{y}^o$ *is an optimal decision, i.e.,* $\boldsymbol{y}^o \in \arg\max_{\boldsymbol{y} \in \mathcal{Y}} r(\boldsymbol{\theta}^*; \boldsymbol{y})$.

A standard assumption for CPE problems is that the optimal decision under the true parameter vector $\boldsymbol{\theta}^*$ is unique, i.e., $\boldsymbol{y}^* = \arg\max_{\boldsymbol{y} \in \mathcal{Y}} r(\boldsymbol{\theta}^*; \boldsymbol{y})$. The performance of a CPE algorithm is measured by its *sample complexity*, which is the number of rounds taken when the algorithm guarantees its output to be the optimal one with probability at least $1 - \delta$.

We say that a reward function $r(\boldsymbol{\theta}; \boldsymbol{y})$ is *continuous* if $r(\boldsymbol{\theta}; \boldsymbol{y})$ is continuous in $\boldsymbol{\theta}$ for every $\boldsymbol{y} \in \mathcal{Y}$, and *(additively) separable* if there exist functions $r_1, \ldots, r_m$ such that $r(\boldsymbol{\theta}; \boldsymbol{y}) = \sum_{i=1}^m r_i(\theta_i, y_i)$. We use CPE-CS to denote the class of CPE problems with Continuous and Separable reward functions and each parameter θ_i^* of arm i can either be mean $\mathbb{E}_{X \sim D_i}[X]$ or variance $\mathrm{Var}_{X \sim D_i}[X]$.[1] We use $\textsc{Est}_i(X_{i,1}, X_{i,2}, \ldots, X_{i,s})$ to denote the unbiased estimator for parameter θ_i^* from s i.i.d. observations $X_{i,1}, X_{i,2}, \ldots, X_{i,s}$ of the i-th arm. In particular, for the mean estimator, $\textsc{Est}_i(X_{i,1}, X_{i,2}, \ldots, X_{i,s}) = \frac{1}{s}\sum_{j=1}^s X_{i,j}$, and for the variance estimator, $\textsc{Est}_i(X_{i,1}, X_{i,2}, \ldots, X_{i,s}) = \frac{1}{s-1}\left(\sum_{j=1}^s X_{i,j}^2 - \frac{1}{s}(\sum_{j=1}^s X_{i,j})^2\right)$. Notice that the variance estimator needs at least two samples. We also define $\phi\colon [0, 1]^m \to \mathcal{Y}$ to be a deterministic tie-breaking maximization oracle such that for any $\boldsymbol{\theta} \in [0, 1]^m$, $\phi(\boldsymbol{\theta}) = (\phi_1(\boldsymbol{\theta}), \ldots, \phi_m(\boldsymbol{\theta})) \in \arg\max_{\boldsymbol{y} \in \mathcal{Y}} r(\boldsymbol{\theta}; \boldsymbol{y})$ and it always outputs the same optimal solution, called the *leading optimal solution*, under the same parameter $\boldsymbol{\theta}$.

CPE-CS encompasses the important CPE problems with Linear reward functions (CPE-L). In CPE-L, parameter θ_i^* is the mean of arm i for each $i \in [m]$. Each decision is a subset of $[m]$, which can be represented as an m-dimensional binary vector. Thus, the decision space $\mathcal{Y}$ is a subset of $\{0, 1\}^m$, and each vector $\boldsymbol{y} = (y_1, \ldots, y_m) \in \mathcal{Y}$ represents a subset

[1] Other parameter θ_i^* of D_i is also acceptable if it has an unbiased estimator from the samples of D_i. Only a minor change is needed in the formula of confidence radius in COCI (Algorithm 1).

Algorithm 1: COCI: Consistently Optimal Confidence Interval Algorithm for CPE-CS

Input: Confidence error bound $\delta \in (0,1)$, maximization oracle ϕ.
Output: $\boldsymbol{y}^o = (y_1, y_2, \ldots, y_m) \in \mathcal{Y}$.

1 $t \leftarrow \tau m$; *// $\tau = 1$ for the mean estimator and $\tau = 2$ for the variance estimator*
2 **for** $i = 1, 2, \ldots, m$ **do**
3 observe the i-th arm τ times $X_{i,1}, \ldots, X_{i,\tau}$;
4 $T_{i,t} \leftarrow \tau$;
5 estimate $\hat{\theta}_{i,t} \leftarrow \textsc{Est}_i(X_{i,1}, \ldots, X_{i,T_{i,t}})$;
6 $\mathrm{rad}_{i,t} \leftarrow \sqrt{\frac{1}{2T_{i,t}} \ln \frac{4t^3}{\tau\delta}}$; *// confidence radius*
7 $\hat{\Theta}_t \leftarrow \{\boldsymbol{\theta} \in [0,1]^m \colon |\theta_i - \hat{\theta}_{i,t}| \le \mathrm{rad}_{i,t}, \forall i \in [m]\}$;
8 **for** $t = \tau m + 1, \tau m + 2, \tau m + 3, \ldots$ **do**
9 $C_t \leftarrow \emptyset$;
10 **for** $i = 1, 2, \ldots, m$ **do**
11 **if** $\max_{\boldsymbol{\theta} \in \hat{\Theta}_{t-1}} \phi_i(\boldsymbol{\theta}) \neq \min_{\boldsymbol{\theta} \in \hat{\Theta}_{t-1}} \phi_i(\boldsymbol{\theta})$ **then**
12 $C_t \leftarrow C_t \cup \{i\}$;
13 **if** $C_t = \emptyset$ **then**
14 **return** $\boldsymbol{y}^o = \phi(\boldsymbol{\theta})$ for an arbitrary $\boldsymbol{\theta} \in \hat{\Theta}_{t-1}$;
15 $j \leftarrow \arg\max_{i \in C_t} \mathrm{rad}_{i,t-1}$;
16 $T_{j,t} \leftarrow T_{j,t-1} + 1$; $T_{i,t} \leftarrow T_{i,t-1}$ for all $i \neq j$;
17 play the j-th arm and observe the outcome $X_{j,T_{j,t}}$;
18 update $\hat{\theta}_{j,t} \leftarrow \textsc{Est}_j(X_{j,1}, \ldots, X_{j,T_{j,t}})$;
19 update $\hat{\theta}_{i,t} \leftarrow \hat{\theta}_{i,t-1}$ for all $i \neq j$;
20 update $\mathrm{rad}_{i,t} \leftarrow \sqrt{\frac{1}{2T_{i,t}} \ln \frac{4t^3}{\tau\delta}}$ for all $i \in [m]$;
21 $\hat{\Theta}_t \leftarrow \{\boldsymbol{\theta} \in [0,1]^m \colon |\theta_i - \hat{\theta}_{i,t}| \le \mathrm{rad}_{i,t}, \forall i \in [m]\}$;

of arms $S_{\boldsymbol{y}} = \{i \in [m] \colon y_i = 1\}$. Moreover, the reward function $r(\boldsymbol{\theta}; \boldsymbol{y}) = \sum_{i=1}^m \theta_i \cdot y_i$ is continuous and separable.

3 Solving CPE-CS

In this section, we propose the Consistently Optimal Confidence Interval (COCI) Algorithm for CPE-CS, and analyze its sample complexity. En route to our sample complexity bound, we introduce a new concept of arm-level *consistently optimal radius* Λ_i of each arm i, which leads to a new hardness measure $\mathbf{H}_{\boldsymbol{\Lambda}}$. We first introduce the components and notations which will be used in the algorithm.

The algorithm we propose for CPE-CS (Algorithm 1) is based on the confidence intervals of the parameter estimates. The algorithm maintains the confidence interval space $\hat{\Theta}_t$ for every round t to guarantee that the true parameter $\boldsymbol{\theta}^*$ is always in $\hat{\Theta}_t$ for all $t > 0$ with probability at least $1 - \delta$. After the initialization (lines 1–7), in each round t, the algorithm first computes the candidate set $C_t \subseteq [m]$ (lines 9–12). According to the key condition in line 11, C_t contains the i-th arm if $\max_{\boldsymbol{\theta} \in \hat{\Theta}_{t-1}} \phi_i(\boldsymbol{\theta}) \neq \min_{\boldsymbol{\theta} \in \hat{\Theta}_{t-1}} \phi_i(\boldsymbol{\theta})$ (this is a logical condition, and its actual implementation will be discussed in Section 3.1). The stopping condition is $C_t = \emptyset$ (line 13), which means that within the confidence interval space, all leading optimal solutions are the same. In this case, the algorithm returns the leading optimal solution under any $\boldsymbol{\theta} \in \hat{\Theta}_{t-1}$ as the final output (line 14). Notice that if the true parameter $\boldsymbol{\theta}^*$ is in $\hat{\Theta}_{t-1}$, then the output is the true optimal solution $\boldsymbol{y}^o = \phi(\boldsymbol{\theta}^*) = \boldsymbol{y}^*$. If $C_t \neq \emptyset$, then the algorithm picks any arm j with the largest confidence radius (line 15), plays this arm, observes its feedback, and updates its estimate $\hat{\theta}_{j,t}$ and confidence radius $\mathrm{rad}_{j,t}$ accordingly (lines 16–21). Intuitively, arm j is the most uncertain arm causing inconsistency, thus the algorithm picks it to play first. Since the key stopping condition is that the leading optimal solutions for all $\boldsymbol{\theta} \in \hat{\Theta}_{t-1}$ are consistently optimal, we call our algorithm Consistently Optimal Confidence Interval (COCI) algorithm.

Before analyzing the sample complexity of the COCI algorithm, we first introduce the (arm-level) *consistent optimality radius* for every arm i, which is formally defined below.

Definition 2. *For all $i \in [m]$, the* consistent optimality radius *Λ_i for arm i is defined as:*

$$\Lambda_i = \inf_{\boldsymbol{\theta} : \phi_i(\boldsymbol{\theta}) \neq \phi_i(\boldsymbol{\theta}^*)} \|\boldsymbol{\theta} - \boldsymbol{\theta}^*\|_\infty .$$

Intuitively, Λ_i measures how far $\boldsymbol{\theta}$ can be away from $\boldsymbol{\theta}^*$ (in infinity norm) while the leading optimal solution under $\boldsymbol{\theta}$ is still consistent with the true optimal one in the i-th dimension, as precisely stated below.

Proposition 1. *$\forall i \in [m]$, if $|\theta_j - \theta_j^*| < \Lambda_i$ holds for all $j \in [m]$, then $\phi_i(\boldsymbol{\theta}) = \phi_i(\boldsymbol{\theta}^*)$.*

The following lemma shows that the consistent optimality radii are all positive, provided by that the reward function is continuous and the true optimal decision $\boldsymbol{y}^*$ is unique.

Lemma 1. *If the reward function $r(\boldsymbol{\theta}; \boldsymbol{y})$ is continuous on $\boldsymbol{\theta}$ for every $\boldsymbol{y} \in \mathcal{Y}$, and the optimal decision $\boldsymbol{y}^*$ under the true parameter vector $\boldsymbol{\theta}^*$ is unique, then Λ_i is positive for every $i \in [m]$.*

Given that the consistent optimality radii are all positive, we can introduce the key hardness measure used in the sample complexity analysis. We define *consistent optimality hardness* as $\mathbf{H}_{\boldsymbol{\Lambda}} = \sum_{i=1}^m \frac{1}{\Lambda_i^2}$. The following theorem shows our primary sample complexity result for the COCI algorithm.

Theorem 1. *With probability at least $1 - \delta$, the COCI algorithm (Algorithm 1) returns the unique true optimal solution $\boldsymbol{y}^o = \boldsymbol{y}^*$, and the number of rounds (or samples) T satisfies*

$$\begin{aligned} T &\le 2m + 12\mathbf{H}_{\boldsymbol{\Lambda}} \ln 24\mathbf{H}_{\boldsymbol{\Lambda}} + 4\mathbf{H}_{\boldsymbol{\Lambda}} \ln \frac{4}{\tau\delta} \\ &= O\left(\mathbf{H}_{\boldsymbol{\Lambda}} \log \frac{\mathbf{H}_{\boldsymbol{\Lambda}}}{\delta}\right). \end{aligned} \tag{1}$$

Theorem 1 shows that the sample complexity is positively related to the consistent optimality hardness, or inversely proportional to the square of consistent optimality radius Λ_i^2. Intuitively, when Λ_i is small, we need more samples to make the optimal solutions in the confidence interval consistent on the i-th dimension, and hence higher sample complexity.

We remark that if we do not compute the candidate set C_t and directly pick the arm with the largest radius among *all* arms in line 15, every arm will be selected in a round-robin

fashion and COCI becomes a uniform sampling algorithm. In the extended version, we show that the sample complexity upper bound of the uniform version is obtained by replacing $\mathbf{H}_{\mathbf{\Lambda}}$ in Eq. (1) by $\mathbf{H}^{\mathrm{U}}_{\mathbf{\Lambda}} = \frac{m}{\min_{i\in[m]}\Lambda_i^2}$, and the factor $\mathbf{H}^{\mathrm{U}}_{\mathbf{\Lambda}}$ is tight for the uniform sampling. This indicates that the adaptive sampling method of COCI would perform much better than the uniform sampling when arms have heterogeneous consistent optimality radii such that $\mathbf{H}_{\mathbf{\Lambda}} \ll \mathbf{H}^{\mathrm{U}}_{\mathbf{\Lambda}}$.

Due to the space constraint, we only provide the key lemma below leading to the proof of the theorem. We define a random event $\xi = \{\forall t \geq \tau m, \forall i \in [m], |\hat{\theta}_{i,t} - \theta^*_i| \leq \mathrm{rad}_{i,t}\}$, which indicates that $\boldsymbol{\theta}^*$ is inside the confidence interval space of all the rounds. Then we have the following lemma.

Lemma 2. *Suppose event ξ occurs. For every $i \in [m]$ and every $t > \tau m$, if $\mathrm{rad}_{i,t-1} < \Lambda_i/2$, then the i-th arm will not be played in round t.*

Proof. Suppose, for a contradiction, that the i-th arm is played in round t, namely, $i \in C_t$, and $i = \arg\max_{j\in C_t} \mathrm{rad}_{j,t-1}$. Thus for each $j \in C_t$, we have $\mathrm{rad}_{j,t-1} \leq \mathrm{rad}_{i,t-1} < \Lambda_i/2$.

We claim that for all $\boldsymbol{\theta} \in \hat{\Theta}_{t-1}$, $\phi_i(\boldsymbol{\theta}) = \phi_i(\boldsymbol{\theta}^*)$. If so, $\max_{\boldsymbol{\theta}\in\hat{\Theta}_{t-1}} \phi_i(\boldsymbol{\theta}) = \min_{\boldsymbol{\theta}\in\hat{\Theta}_{t-1}} \phi_i(\boldsymbol{\theta})$, then by line 11 $i \notin C_t$, a contradiction.

We now prove the claim. For any vector $\boldsymbol{x} \in \mathbb{R}^m$ and any index subset $C \subseteq [m]$, we use $\boldsymbol{x}_C$ to denote the subvector of $\boldsymbol{x}$ projected onto C. For vector-valued functions such as $\phi(\boldsymbol{\theta})$, we use $\phi_C(\boldsymbol{\theta})$ for $\phi(\boldsymbol{\theta})_C$. For any $\boldsymbol{\theta} \in \hat{\Theta}_{t-1}$, we construct an intermediate vector $\boldsymbol{\theta}' = (\boldsymbol{\theta}_{C_t}, \boldsymbol{\theta}^*_{-C_t})$, i.e., the j-th component θ'_j is θ_j when $j \in C_t$, or θ^*_j when $j \notin C_t$. Since event ξ occurs, we have $|\hat{\theta}_{j,t-1} - \theta^*_j| \leq \mathrm{rad}_{j,t-1}$ for $j \in [m]$. Thus for all $j \in C_t$, $|\theta'_j - \theta^*_j| \leq |\theta_j - \hat{\theta}_{j,t-1}| + |\hat{\theta}_{j,t-1} - \theta^*_j| \leq 2\,\mathrm{rad}_{j,t-1} < \Lambda_i$, and for all $j \notin C_t$, $|\theta'_j - \theta^*_j| = 0$. This means that $\|\boldsymbol{\theta}' - \boldsymbol{\theta}^*\|_\infty < \Lambda_i$. According to Proposition 1, $\phi_i(\boldsymbol{\theta}') = \phi_i(\boldsymbol{\theta}^*)$. We next prove that $\phi_i(\boldsymbol{\theta}) = \phi_i(\boldsymbol{\theta}')$, which directly leads to $\phi_i(\boldsymbol{\theta}) = \phi_i(\boldsymbol{\theta}^*)$.

Since event ξ occurs and $\boldsymbol{\theta}^* \in [0,1]^m$, $\boldsymbol{\theta}^*$ is in $\hat{\Theta}_{t-1}$. By the definition of $\boldsymbol{\theta}'$ and $\boldsymbol{\theta} \in \hat{\Theta}_{t-1}$, $\boldsymbol{\theta}'$ is also in $\hat{\Theta}_{t-1}$. According to Algorithm 1, for each $j \notin C_t$, we have $\max_{\boldsymbol{\theta}\in\hat{\Theta}_{t-1}} \phi_j(\boldsymbol{\theta}) = \min_{\boldsymbol{\theta}\in\hat{\Theta}_{t-1}} \phi_j(\boldsymbol{\theta})$, thus $\phi_{-C_t}(\boldsymbol{\theta}) = \phi_{-C_t}(\boldsymbol{\theta}') = \phi_{-C_t}(\boldsymbol{\theta}^*)$.

Note that the reward function is separable, we have

$$r(\boldsymbol{\theta}; \boldsymbol{y}) = \sum_{j\in C_t} r_j(\theta_j, y_j) + \sum_{j\notin C_t} r_j(\theta_j, y_j).$$

Let $\mathcal{Y}_{C_t}(\boldsymbol{\theta}) = \{\boldsymbol{y}_{C_t} \colon \boldsymbol{y} \in \mathcal{Y} \wedge \boldsymbol{y}_{-C_t} = \phi_{-C_t}(\boldsymbol{\theta})\}$. It is straightforward to verify that $\phi_{C_t}(\boldsymbol{\theta})$ is the leading optimal solution for the following problem:

$$\begin{aligned} \max \quad & \sum_{j\in C_t} r_j(\theta_j, z_j), \\ \text{subject to} \quad & \boldsymbol{z} \in \mathcal{Y}_{C_t}(\boldsymbol{\theta}). \end{aligned} \tag{2}$$

Similarly, we have

$$r(\boldsymbol{\theta}'; \boldsymbol{y}) = \sum_{j\in C_t} r_j(\theta_j, y_j) + \sum_{j\notin C_t} r_j(\theta^*_j, y_j),$$

and $\phi_{C_t}(\boldsymbol{\theta}')$ is the leading optimal solution for

$$\begin{aligned} \max \quad & \sum_{j\in C_t} r_j(\theta_j, z_j), \\ \text{subject to} \quad & \boldsymbol{z} \in \mathcal{Y}_{C_t}(\boldsymbol{\theta}^*). \end{aligned} \tag{3}$$

Since $\phi_{-C_t}(\boldsymbol{\theta}) = \phi_{-C_t}(\boldsymbol{\theta}^*)$, optimization problems (2) and (3) are identical, thus they have the some leading optimal solution $\phi_{C_t}(\boldsymbol{\theta}) = \phi_{C_t}(\boldsymbol{\theta}')$. Notice that $i \in C_t$, therefore, $\phi_i(\boldsymbol{\theta}) = \phi_i(\boldsymbol{\theta}')$ holds. □

The above lemma is the key connecting consistent optimality radius Λ_i with confidence radius $\mathrm{rad}_{i,t-1}$ and the stopping condition. Its proof relies on both the definition of consistent optimality radius and the assumption of separable reward functions. With this lemma, the sample complexity can be obtained by considering the first round when every arm satisfies the condition $\mathrm{rad}_{i,t-1} < \Lambda_i/2$.

Borrowing a lower bound analysis in [Chen *et al.*, 2017], we can further show that the hardness measure $\mathbf{H}_{\mathbf{\Lambda}}$ is necessary for CPE-CS, even CPE-L, as shown below.

Theorem 2. *Given m arms and $\delta \in (0, 0.1)$, there exists an instance such that every algorithm $\mathcal{A}$ for CPE-L which outputs the optimal solution with probability at least $1-\delta$, takes at least*

$$\Omega(\mathbf{H}_{\mathbf{\Lambda}} + \mathbf{H}_{\mathbf{\Lambda}} m^{-1} \log \delta^{-1})$$

samples in expectation.

3.1 Implementing the Condition in Line 11

The key condition in line 11 of Algorithm 1 is a logical one revealing the conceptual meaning of the stopping condition, but it does not lead to a direct implementation. In many CPE-CS instances, the condition can be translated to a condition only on the boundary of $\hat{\Theta}_{t-1}$, and further due to the bi-monotonicity of ϕ introduced below, it has an efficient implementation. Such instances include best-arm identification, top-k arm identification, water resources planning (Section 4.1), partitioned opinion sampling (Section 4.2), etc.

We say that the leading optimal solution $\phi(\boldsymbol{\theta})$ satisfies *bi-monotonicity*, if for each $i \in [m]$, $\phi_i(\boldsymbol{\theta})$ is monotonically non-increasing (or non-decreasing) in θ_i, and monotonically non-decreasing (or non-increasing) in θ_j for all $j \neq i$. For convenience, we use $\overline{\theta}_{i,t} = \max_{\boldsymbol{\theta}\in\hat{\Theta}_t} \theta_i$ and $\underline{\theta}_{i,t} = \min_{\boldsymbol{\theta}\in\hat{\Theta}_t} \theta_i$ to denote the upper and lower confidence bound of arm i in round t. We also use $\overline{\boldsymbol{\theta}}_{-i,t}$ and $\underline{\boldsymbol{\theta}}_{-i,t}$ to denote the upper and lower confidence bounds of all arms excluding arm i.

Theorem 3. *If the leading optimal solution $\phi(\boldsymbol{\theta})$ satisfies bi-monotonicity, the condition in line 11 of Algorithm 1 can be efficiently implemented by*

$$\phi_i(\underline{\boldsymbol{\theta}}_{-i,t-1}, \overline{\theta}_{i,t-1}) \neq \phi_i(\overline{\boldsymbol{\theta}}_{-i,t-1}, \underline{\theta}_{i,t-1}).$$

The above theorem indicates that, when bi-monotonicity holds for $\phi(\boldsymbol{\theta})$, we only need two calls to the offline oracle $\phi(\boldsymbol{\theta})$ to implement the condition in line 11, and thus the COCI algorithm has an efficient implementation in this case.

4 Applications

4.1 Water Resource Planning

Water resource systems benefit people to meet drinking water and sanitation needs, and also support and maintain resilient biodiverse ecosystems. In regional water resource planning, one need to determine the Biological Oxygen Demand (BOD, a measure of pollution) to be removed from the water system at each source. Online learning techniques proposed in recent years make adaptive optimization for water resource planning possible.

Let y_i be the pounds of BOD to be removed at source i. One general model (adapted from [Bradley *et al.*, 1977]) to minimize total costs to the region to meet specified pollution standards can be expressed as:

$$\begin{aligned} \max \quad & \sum_{i=1}^{m} \theta_i^* y_i - \sum_{i=1}^{m} f_i(y_i), \\ \text{subject to} \quad & \sum_{i=1}^{m} y_i \geq b, 0 \leq y_i \leq c_i, \forall i \in [m], \end{aligned} \tag{4}$$

where θ_i^* is the quality response caused by removing one pound of BOD at source i, and $f_i(y_i)$ is the cost of removing y_i pounds of BOD at source i. Each y_i is constrained by c_i, the maximum pounds of BOD that can be removed at source i. Moreover, the total pounds of BOD to be removed are required to be larger than a certain threshold b.

The above model formulates the trade-off between the benefit and the cost of removing the pollutants. The cost function f_i is usually known and non-linear, which may depend on the cost of oxidation, labor cost, facility cost, etc., while the quality response θ_i^* is unknown beforehand, and needs to be learned from tests at source i. In each test, the tester measures the quality response at a source i and gets an observation of θ_i^*, which can be regarded as a random variable θ_i derived from an unknown distribution with mean θ_i^*. The goal is to do as few tests as possible to estimate the quality responses, and then give a final allocation $(y_1^o, \ldots, y_m^o)$ of BOD among sources as the plan to be implemented (e.g., building BOD removal facilities at the sources).

The above problem falls into the CPE-CS framework. The i-th source corresponds to the i-th arm. Each quality response at source i is the unknown parameter θ_i^* associated with the arm i, and $\tau = 1$. Each allocation $(y_1, \ldots, y_m)$ satisfying the constraints corresponds to a decision. We discretize $\{y_i\}$'s so that the decision class $\mathcal{Y}$ is finite. The reward function is $r(\boldsymbol{\theta}, \boldsymbol{y}) = \sum_{i=1}^{m} \theta_i y_i - \sum_{i=1}^{m} f_i(y_i)$, which is continuous and separable. Suppose the offline problem of Eq. (4) when $\boldsymbol{\theta}^*$ is known can be solved by a known oracle $\phi(\boldsymbol{\theta}^*)$. Then, the COCI algorithm can be directly applied to the water resource planning problem. The following lemma gives a sufficient condition for the bi-monotonicity of ϕ.

Lemma 3. *When $\{\mathrm{d}f_i/\mathrm{d}y_i\}$'s are all monotonically increasing or decreasing, and the constraint $\sum_{i=1}^{m} y_i \geq b$ is tight at the leading optimal solution $\phi(\boldsymbol{\theta})$ for all $\boldsymbol{\theta}$, then $\phi(\boldsymbol{\theta})$ satisfies bi-monotonicity.*

By Theorem 3, when the offline oracle for the water resources planning problem satisfies bi-monotonicity, we can instantiate the condition in line 11 of Algorithm 1 as $\phi_i(\underline{\boldsymbol{\theta}}_{-i,t-1}, \overline{\theta}_{i,t-1}) \neq \phi_i(\overline{\boldsymbol{\theta}}_{-i,t-1}, \underline{\theta}_{i,t-1})$.

Although this application is set up in the context of water resource planning, we can see that the formulation in Eq. (4) is general enough to model other applications, especially ones in the urban planning context. For example, for planning air quality control for a city, we need to target a number of air pollution emission sources (e.g., factories), and do adaptive testing at the sources to determine the optimal pollution remove target at each sources which maximizes the total utility of the planning. Other applications, such as crime control, may also be modeled similarly as instances of our CPE-CS framework and solved effectively by our COCI algorithm.

4.2 Partitioned Opinion Sampling

Public opinion dynamics has been well studied, and there are a number of opinion dynamic models proposed in the literature, such as the voter model [Clifford and Sudbury, 1973], and its variants [Yildiz *et al.*, 2011; Li *et al.*, 2015; Huang *et al.*, 2017]. In these models, people's opinions $f_1^{(t)}, f_2^{(t)}, \ldots, f_n^{(t)} \in [0, 1]$ change over time t, and will converge to a steady state after sufficient social interactions in which the joint distribution of people's opinions no longer changes. Thus, they are regarded as Bernoulli random variables derived from the steady-state joint distribution, and sampling at time t can be considered as observing part of a realization of $f_1^{(t)}, f_2^{(t)}, \ldots, f_n^{(t)}$. In partitioned opinion sampling, the population is divided into several disjoint groups $V_1, V_2, \ldots, V_m$ with $n_i = |V_i|$. When we draw y_i samples (with replacement) from group V_i at time t, we obtain y_i i.i.d. random variables $f_{v_{i,1}}^{(t)}, f_{v_{i,2}}^{(t)}, \ldots, f_{v_{i,y_i}}^{(t)}$, where $v_{i,j}$ is the j-th sample from group V_i. Partitioned sampling uses $\hat{f}^{(t)} = \sum_{i=1}^{m} \frac{n_i}{n} \cdot \left(\frac{1}{y_i} \sum_{j=1}^{y_i} f_{v_{i,j}}^{(t)} \right)$ as the unbiased estimator for the mean population opinion at time t, and the task is to find the optimal allocation $(y_1^o, \ldots, y_m^o)$ with sample size budget $\sum_{i=1}^{m} y_i^o \leq k$ which minimizes the sample variance $\mathrm{Var}[\hat{f}^{(t)}]$, a common sample quality measure [Bethel, 1986; Ballin and Barcaroli, 2013; Huang *et al.*, 2017].

One way to achieve best estimate quality for a future time t is to do adaptive sampling to quickly estimate the opinion variance of each group, and then decide the optimal sample size allocation for the real sample event at time t. This corresponds to certain opinion polling practices, for instance, polling after each presidential debates, and preparing for a better sample quality at the election day. We remark that in this setting, past samples are useful to estimate opinion variance within groups, but cannot be directly use to estimate the mean opinion at a future time t, since $\hat{f}^{(t)}$ is time-based and using historical samples directly may lead to biased estimates.

More specifically, let X_i be the result of one random sample from group V_i in the steady state. Note that the randomness of X_i comes from both the sampling randomness and the opinion randomness in the steady state. One can easily verify that $\mathrm{Var}[\hat{f}^{(t)}] = \sum_{i=1}^{m} \frac{n_i^2}{n^2 y_i} \mathrm{Var}[X_i]$, where $\mathrm{Var}[X_i]$ is the variance of group V_i, and referred to as the *within-group variance*. The goal is to use as few samples as possible to estimate within-group variances, and then give the final sample

size allocation which minimizes $\mathrm{Var}[\hat{f}^{(t)}]$.

This falls into the CPE-CS framework. In particular, each group V_i corresponds to an arm i, and each within-group variance $\mathrm{Var}[X_i]$ corresponds to the unknown parameter θ^*_i of arm i. The decision space $\mathcal{Y}$ is $\{(y_1, \ldots, y_m) \in \mathbb{Z}^m_+ \colon \sum_{i=1}^m y_i \le k\}$. The reward function $r(\boldsymbol{\theta}; \boldsymbol{y})$ is set to be $-\sum_{i=1}^m \frac{n_i^2 \theta_i}{n^2 y_i}$, where the negative sign is because the partitioned opinion sampling problem is a minimization problem. It is non-linear but continuous and separable. Therefore, the problem is an instance of CPE-CS. The oracle for the offline problem can be achieved by a greedy algorithm, denoted as $\phi(\boldsymbol{\theta})$, and it satisfies the bi-monotonicity (the design and the analysis of the offline oracle is non-trivial, see the extended version). Thus, the COCI algorithm can be directly applied as follows: 1) $\textsc{Est}_i$ is set to be the variance estimator, i.e., $\textsc{Est}_i(X_{i,1}, \ldots, X_{i,s}) = \frac{1}{s-1}(\sum_{j=1}^s X_{i,j}^2 - \frac{1}{s}(\sum_{j=1}^s X_{i,j})^2)$, and $\tau = 2$; 2) the condition in line 11 of Algorithm 1 is instantiated by $\phi_i(\underline{\boldsymbol{\theta}}_{-i,t-1}, \overline{\theta}_{i,t-1}) \neq \phi_i(\overline{\boldsymbol{\theta}}_{-i,t-1}, \underline{\theta}_{i,t-1})$.

5 Applying COCI to CPE-L

In Section 2, we already show that the linear class CPE-L is a special case of CPE-CS. In this section, we discuss the implication of applying COCI algorithm to solve CPE-L problems, and compare the sample complexity and implementation efficiency against the CLUCB algorithm in [Chen *et al.*, 2014]. Since the parameter $\boldsymbol{\theta}^*$ is the vector of means of arms, we use the mean estimator and set $\tau = 1$ in COCI.

Recall that for a binary vector $\boldsymbol{y} \in \mathcal{Y}$, $S_{\boldsymbol{y}}$ is defined as $\{i \in [m] \colon y_i = 1\}$. Chen *et al.* [2014] use the term *reward gap* in the formulation of sample complexity. For each arm $i \in [m]$, its *reward gap* Δ_i is defined as:

$$\Delta_i = \begin{cases} r(\boldsymbol{\theta}^*; \boldsymbol{y}^*) - \max_{\boldsymbol{y} \in \mathcal{Y}, i \notin S_{\boldsymbol{y}}} r(\boldsymbol{\theta}^*; \boldsymbol{y}), \text{if } i \in S_{\boldsymbol{y}^*}, \\ r(\boldsymbol{\theta}^*; \boldsymbol{y}^*) - \max_{\boldsymbol{y} \in \mathcal{Y}, i \in S_{\boldsymbol{y}}} r(\boldsymbol{\theta}^*; \boldsymbol{y}), \text{if } i \notin S_{\boldsymbol{y}^*}. \end{cases}$$

Chen *et al.* [2014] also define a (reward gap) hardness measure $\mathbf{H}_{\boldsymbol{\Delta}} = \sum_{i=1}^m \frac{1}{\Delta_i^2}$. Moreover, for each decision class $\mathcal{Y}$, Chen *et al.* [2014] define a key quantity *width*, denoted as $\mathrm{width}(\mathcal{Y})$, that is needed for sample complexity. Intuitively, $\mathrm{width}(\mathcal{Y})$ denotes the minimum number of elements that one may need to exchange in one step of a series of steps when changing the current decision $S \in \mathcal{Y}$ into another decision $S' \in \mathcal{Y}$, and for every step of exchange in the series, the resulting decision (subset) should still be in $\mathcal{Y}$. The technical definition is not very relevant with the discussion below, and thus is left in the supplementary material. We remark that $\mathrm{width}(\mathcal{Y}) = O(m)$.

Given the above setup, Chen *et al.* [2014] show that with probability $1 - \delta$, their CLUCB algorithm achieves sample complexity bound

$$\begin{aligned} T &\le 2m + 499\mathrm{width}(\mathcal{Y})^2 \mathbf{H}_{\boldsymbol{\Delta}} \ln(4m\mathrm{width}(\mathcal{Y})^2 \mathbf{H}_{\boldsymbol{\Delta}}/\delta) \\ &= O\left(\mathrm{width}(\mathcal{Y})^2 \mathbf{H}_{\boldsymbol{\Delta}} \log(m\mathbf{H}_{\boldsymbol{\Delta}}/\delta)\right). \end{aligned} \tag{5}$$

When applying the COCI algorithm to solve CPE-L problems, we are able to obtain the following key connection between consistent optimality radius and the reward gap:

Lemma 4. *For the CPE-L problems, we have* $\forall i \in [m]$, $\Lambda_i \ge \Delta_i/\mathrm{width}(\mathcal{Y})$, *and thus* $\mathbf{H}_{\boldsymbol{\Lambda}} \le \mathbf{H}_{\boldsymbol{\Delta}} \cdot \mathrm{width}(\mathcal{Y})^2$.

Combining with Theorem 1, we have that COCI could achieve the following sample complexity bound for CPE-L:

$$\begin{aligned} T &\le 2m + 12\mathrm{width}(\mathcal{Y})^2 \mathbf{H}_{\boldsymbol{\Delta}} \ln(24\mathrm{width}(\mathcal{Y})^2 \mathbf{H}_{\boldsymbol{\Delta}}) \\ &\quad + 4\mathrm{width}(\mathcal{Y})^2 \mathbf{H}_{\boldsymbol{\Delta}} \ln(4\delta^{-1}) \\ &= O\left(\mathrm{width}(\mathcal{Y})^2 \mathbf{H}_{\boldsymbol{\Delta}} \log(m\mathbf{H}_{\boldsymbol{\Delta}}/\delta)\right). \end{aligned}$$

The above result has the same sample complexity[2] as in Eq. (5) (with even a slightly better constant). However, with our analysis, we only need the complicated combinatorial quantity $\mathrm{width}(\mathcal{Y})$ and the linear reward assumption in the last step. This also suggests that our consistent optimality radius Λ_i and its associated consistent optimality hardness $\mathbf{H}_{\boldsymbol{\Lambda}}$ are more fundamental measures of problem hardness than the reward gap Δ_i and its associated reward gap hardness $\mathbf{H}_{\boldsymbol{\Delta}}$.

Next we discuss the implementation of the condition in line 11 of COCI for CPE-L. First, because linear functions are monotone, it is easy to see that we only need to check parameters $\boldsymbol{\theta}$ on the boundaries of $\hat{\Theta}_{t-1}$ (at most $2|\mathcal{Y}|$ calls to the oracle ϕ). For simple constraints such as any subsets of size k, it is easy to verify that $\phi(\boldsymbol{\theta})$ is bi-monotone in this case, and thus we have efficient implementation of the condition as given in Theorem 3. For more complicated combinatorial constraints, it is still an open question on whether efficient implementation of the condition in line 11 exists when oracle ϕ is given. The CLUCB algorithm, on the other hand, does have an efficient implementation for all CPE-L problems as long as the oracle ϕ is given.

Therefore, compared with CLUCB in terms of efficient implementation, COCI can be viewed as taking the trade-off between the complexity of the reward functions and the complexity of combinatorial constraints. In particular, COCI could handle more complicated nonlinear reward functions on real vectors, and allow efficient implementation (due to bi-monotonicity) under simple constraints, while CLUCB deals with complicated combinatorial constraints but could only work with linear reward functions on binary vectors.

6 Future Work

There are a number of open problems and future directions. For example, one can consider the fixed budget setting of CPE-CS: the game stops after a fixed number T of rounds where T is given before the game starts, and the learner needs to minimize the probability of error $\Pr[\boldsymbol{y}^o \neq \boldsymbol{y}^*]$. One may also consider the PAC setting: with probability at least $1 - \delta$ the algorithm should output a decision with reward at most ε away from the optimal reward. This setting may further help to eliminate the requirement of finite decision class $\mathcal{Y}$. Another direction is to combine the advantage of COCI and CLUCB to design a unified algorithm that allows efficient implementation for all CPE-CS problems. How to incorporate approximation oracle instead of the exact oracle into the CPE framework is also an interesting direction.

[2] CPE-L in [Chen *et al.*, 2014] assumes R-sub-Gaussian distributions. Our analysis can be adapted to R-sub-Gaussian distributions as well, with the same R^2 term appearing in the sample complexity.

Acknowledgments

This work was supported in part by the National Basic Research Program of China Grant 2011CBA00300, 2011CBA00301, the National Natural Science Foundation of China Grant 61033001, 61361136003, 61433014.

References

[Audibert and Bubeck, 2010] Jean-Yves Audibert and Sébastien Bubeck. Best arm identification in multi-armed bandits. In *COLT*, pages 41–53, 2010.

[Auer *et al.*, 2002a] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3):235–256, 2002.

[Auer *et al.*, 2002b] Peter Auer, Nicolo Cesa-Bianchi, Yoav Freund, and Robert E Schapire. The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1):48–77, 2002.

[Ballin and Barcaroli, 2013] Marco Ballin and Giulio Barcaroli. Joint determination of optimal stratification and sample allocation using genetic algorithm. *Survey Methodology*, 39(2):369–393, 2013.

[Bethel, 1986] James William Bethel. *An optimum allocation algorithm for multivariate surveys*. 1986.

[Bradley *et al.*, 1977] Stephen Bradley, Arnoldo Hax, and Thomas Magnanti. Applied mathematical programming. 1977.

[Bubeck *et al.*, 2011] Sébastien Bubeck, Rémi Munos, and Gilles Stoltz. Pure exploration in finitely-armed and continuous-armed bandits. *Theoretical Computer Science*, 412(19):1832–1852, 2011.

[Bubeck *et al.*, 2012] Sébastien Bubeck, Nicolo Cesa-Bianchi, et al. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012.

[Bubeck *et al.*, 2013] Séebastian Bubeck, Tengyao Wang, and Nitin Viswanathan. Multiple identifications in multi-armed bandits. In *ICML*, pages 258–265, 2013.

[Cesa-Bianchi and Lugosi, 2012] Nicolo Cesa-Bianchi and Gábor Lugosi. Combinatorial bandits. *Journal of Computer and System Sciences*, 78(5):1404–1422, 2012.

[Chen *et al.*, 2014] Shouyuan Chen, Tian Lin, Irwin King, Michael R Lyu, and Wei Chen. Combinatorial pure exploration of multi-armed bandits. In *NIPS*, pages 379–387, 2014.

[Chen *et al.*, 2016a] Lijie Chen, Anupam Gupta, and Jian Li. Pure exploration of multi-armed bandit under matroid constraints. In *COLT*, pages 647–669, 2016.

[Chen *et al.*, 2016b] Wei Chen, Wei Hu, Fu Li, Jian Li, Yu Liu, and Pinyan Lu. Combinatorial multi-armed bandit with general reward functions. In *NIPS*, pages 1659–1667, 2016.

[Chen *et al.*, 2016c] Wei Chen, Yajun Wang, Yang Yuan, and Qinshi Wang. Combinatorial multi-armed bandit and its extension to probabilistically triggered arms. *The Journal of Machine Learning Research*, 17(1):1746–1778, 2016.

[Chen *et al.*, 2017] Lijie Chen, Anupam Gupta, Jian Li, Mingda Qiao, and Ruosong Wang. Nearly optimal sampling algorithms for combinatorial pure exploration. *arXiv:1706.01081*, 2017.

[Clifford and Sudbury, 1973] Peter Clifford and Aidan Sudbury. A model for spatial conflict. *Biometrika*, 60(3):581–588, 1973.

[Combes *et al.*, 2015] Richard Combes, Mohammad Sadegh Talebi Mazraeh Shahi, Alexandre Proutiere, et al. Combinatorial bandits revisited. In *NIPS*, pages 2116–2124, 2015.

[Gabillon *et al.*, 2011] Victor Gabillon, Mohammad Ghavamzadeh, Alessandro Lazaric, and Sébastien Bubeck. Multi-bandit best arm identification. In *NIPS*, pages 2222–2230, 2011.

[Gabillon *et al.*, 2012] Victor Gabillon, Mohammad Ghavamzadeh, and Alessandro Lazaric. Best arm identification: A unified approach to fixed budget and fixed confidence. In *NIPS*, pages 3212–3220, 2012.

[Gai *et al.*, 2012] Yi Gai, Bhaskar Krishnamachari, and Rahul Jain. Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations. *IEEE/ACM Transactions on Networking (TON)*, 20(5):1466–1478, 2012.

[Gopalan *et al.*, 2014] Aditya Gopalan, Shie Mannor, and Yishay Mansour. Thompson sampling for complex online problems. In *ICML*, pages 100–108, 2014.

[Huang *et al.*, 2017] Weiran Huang, Liang Li, and Wei Chen. Partitioned sampling of public opinions based on their social dynamics. In *AAAI*, pages 24–30, 2017.

[Huang *et al.*, 2018] Weiran Huang, Jungseul Ok, Liang Li, and Wei Chen. Combinatorial pure exploration with continuous and separable reward functions and its applications (extended version). *arXiv:1805.01685*, 2018.

[Kalyanakrishnan and Stone, 2010] Shivaram Kalyanakrishnan and Peter Stone. Efficient selection of multiple bandit arms: Theory and practice. In *ICML*, pages 511–518, 2010.

[Kalyanakrishnan *et al.*, 2012] Shivaram Kalyanakrishnan, Ambuj Tewari, Peter Auer, and Peter Stone. PAC subset selection in stochastic multi-armed bandits. In *ICML*, volume 12, pages 655–662, 2012.

[Kaufmann and Kalyanakrishnan, 2013] Emilie Kaufmann and Shivaram Kalyanakrishnan. Information complexity in bandit subset selection. In *COLT*, pages 228–251, 2013.

[Kveton *et al.*, 2014] Branislav Kveton, Zheng Wen, Azin Ashkan, Hoda Eydgahi, and Brian Eriksson. Matroid bandits: Fast combinatorial optimization with learning. In *Proceedings of the Thirtieth Conference on Uncertainty in Artificial Intelligence*, UAI'14, pages 420–429, 2014.

[Kveton *et al.*, 2015] Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvari. Tight regret bounds for stochastic combinatorial semi-bandits. In *Artificial Intelligence and Statistics*, pages 535–543, 2015.

[Lai and Robbins, 1985] Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in applied mathematics*, 6(1):4–22, 1985.

[Li *et al.*, 2015] Yanhua Li, Wei Chen, Yajun Wang, and Zhi-Li Zhang. Voter model on signed social networks. *Internet Mathematics*, 11(2):93–133, 2015.

[Soare *et al.*, 2014] Marta Soare, Alessandro Lazaric, and Rémi Munos. Best-arm identification in linear bandits. In *NIPS*, pages 828–836, 2014.

[Yildiz *et al.*, 2011] Ercan Yildiz, Daron Acemoglu, Asuman E Ozdaglar, Amin Saberi, and Anna Scaglione. Discrete opinion dynamics with stubborn agents. 2011.

[Zhou *et al.*, 2014] Yuan Zhou, Xi Chen, and Jian Li. Optimal PAC multiple arm identification with applications to crowdsourcing. In *ICML*, pages 217–225, 2014.