Front.Comput.Sci.
DOI

RESEARCH ARTICLE

Follow the Perturbed Approximate Leader for Solving Semi-bandit Combinatorial Optimization

Feidiao YANG [1,2], **Wei CHEN** [3], **Jialin ZHANG** [1,2], **Xiaoming SUN** (✉)[1,2]

1 Institute of Computing Technology, Chinese Academy of Sciences, Bejing 100190, China
2 University of Chinese Academy of Sciences, Beijing 100190, China
3 Microsoft Research Asia, Beijing 100080, China

Abstract Combinatorial optimization in the face of uncertainty is a challenge in both operational research and machine learning. In this paper, we consider a special and important class called the adversarial online combinatorial optimization with semi-bandit feedback, in which a player makes combinatorial decisions and gets the corresponding feedback repeatedly. While existing algorithms focus on the regret guarantee or assume there exists an efficient offline oracle, it is still a challenge to solve this problem efficiently if the offline counterpart is NP-hard. In this paper, we propose a variant of the Follow-the-Perturbed-Leader (FPL) algorithm to solve this problem. Unlike the existing FPL approach, our method employs an approximation algorithm as an offline oracle and perturbs the collected data by adding nonnegative random variables. Our approach is simple and computationally efficient. Moreover, it can guarantee a sublinear $(1+\varepsilon)$-scaled regret of order $O(T^{\frac{2}{3}})$ for any small $\varepsilon > 0$ for an important class of combinatorial optimization problems that admit an FPTAS (fully polynomial time approximation scheme), in which T is the number of rounds of the learning process. In addition to the theoretical analysis, we also conduct a series of experiments to demonstrate the performance of our algorithm.

Keywords online learning, online combinatorial optimization, semi-bandit, follow-the-perturbed-leader

Received month dd, yyyy; accepted month dd, yyyy

E-mail: sunxiaoming@ict.ac.cn

1 Introduction

1.1 Background

Combinatorial optimization in the face of uncertainty is challenging in both operational research and machine learning. Conventional combinatorial optimization assumes that the information of a problem instance is completely known and the goal is to optimize some objective. However, this assumption does not always hold in many applications. In such cases, some information on a problem instance is only available after an action is taken. For example, in the path planning problem of minimizing the travel time, the actual passing time of a path is uncertain, depending on various factors such as the traffic condition, the driving skill, and even the weather. It is only available after the actual pass. Therefore, it is interesting and valuable to study the problem of combinatorial optimization in environments with uncertainty.

Online combinatorial optimization is an important problem in such a topic. It models the problem of sequential decision making with combinatorial constraints without knowledge of the future. More specifically, in online combinatorial optimization, a player repeatedly takes actions and observes the corresponding feedback information at the end of each round. This model is especially useful in real-world applications, such as planning and online recommendation.

There are different variants of the online combinatorial optimization problem. In the stochastic setting, problem instances are i.i.d. samples from a stationary distribution, while they might be designated by an adversary in an arbitrary way

in the adversarial setting. In terms of the feedback information, in the full information setting the player can observe the whole information about the problem instance at the end of each round, while only the total loss is available in the (full) bandit setting.

In this paper, we focus on a special case, the adversarial combinatorial semi-bandit problem. In this setting, the player can observe more detailed partial feedback information based on its own decision. Specifically, in addition to the total loss as in the bandit setting, the player can observe the individual cost of each element he selected. Besides, we make no assumption on the problem instance generation and they can be designated by an adversary in an arbitrary way. In other words, we assume the adversary can get access to the player's strategy, except the random numbers it uses (if there is any). Particularly, we consider the adaptive adversarial setting that a problem instance is allowed to depend on the previous actions by the player. The adversarial semi-bandit setting is closer to the scenarios than the other settings in many applications in which the availability of full information or full-bandit information is too extreme and the stochastic assumption does not always hold.

Various algorithms have been proposed for the adversarial combinatorial semi-bandit problem. However, most of them, such as the classical Exponential Weighted Average (EWA) method [1] and the emerging Online Stochastic Mirror Descent (OSMD) approach [2], focus on the regret guarantee and neglect the computational complexity. Recently, Neu and Bartók [3, 4] consider the computational efficiency issue and introduce the Follow-the-Perturbed-Leader (FPL) method [5] to solve the adversarial combinatorial semi-bandit problem. However, Neu and Bartók's method only works well as long as the offline counterpart of the online problem has an exact polynomial-time algorithm. It is still a challenge to solve the adversarial semi-bandit problem efficiently if the underlying combinatorial optimization problem is NP-hard.

1.2 Our Contribution

The basic idea of the FPL method is to reduce an online problem to its offline version such that the well-studied offline combinatorial optimization algorithms could be utilized. In addition, for standing against the adversary, the FPL method perturbs the collected data with random noise, particularly, by subtracting exponentially distributed random variables.

However, the FPL method actually cannot guarantee the computational efficiency if the offline problem is NP-hard since it calls an inefficient offline oracle in each round. Given that efficient approximation algorithms have been developed for many NP-hard problems, in this paper we propose an idea of employing an approximation algorithm as an offline oracle to replace the exact optimization oracle to solve the adversarial combinatorial semi-bandit problem. Our approach is simple, efficient, and it can provide a good performance guarantee for some important cases. Specifically, we summarize our work as follows.

- We design an algorithm that utilizes an offline approximation oracle to solve the adversarial combinatorial semi-bandit problem, which leads to a computationally efficient approach.
- Unlike the original FPL method, our approach perturbs the collected data by adding nonnegative uniformly distributed random variables. The major reason is for adapting to the approximation oracle since many approximation algorithms only accept nonnegative inputs and the original method may fail in this case.
- Our algorithm is computationally efficient. The expected number of calls to the oracle in each round is not greater than the offline problem size and the expected total running time is polynomial to the entire problem size.
- We show that for an important class of combinatorial optimization problems that admit an FPTAS, our approach can guarantee a sublinear $(1+\varepsilon)$-scaled regret of order $O(T^{\frac{2}{3}})$ for any $\varepsilon \in (0, 2]$, where T is the number of rounds of the bandit game. This result is significant since some important combinatorial optimization problems, such as the knapsack problem and some special scheduling problems, admit an FPTAS, and many real-world applications could be reduced to these typical problems. The analysis is challenging since some techniques suitable for exact optimization oracles do not work for approximation oracles. Therefore, we made some key and necessary modification to some existing work to adapt to approximation oracles.
- We conduct a series of experiments to demonstrate the performance of our algorithm. Seeing from the experimental results, our method runs much faster than that with an exact oracle. In terms of regret, our algorithm performs closely to the FPL method with an exact oracle if the problem admits an FPTAS, and it outperforms the CUCB-like algorithms in non-stochastic adversarial settings even in general cases.

1.3 Related Work

The simplest case of the combinatorial bandit problem is the standard multi-armed bandit model where a single arm is to be selected and only the corresponding information is provided in each round. It is an important topic in machine learning and has many applications [6]. Its stochastic setting can date back to Thompson [7] and Robbins [8]. One of the well-studied and widely-used algorithms is the Upper Confidence Bound (UCB) approach proposed by Auer et al. [9]. The adversarial multi-armed bandit and the corresponding Exp3 algorithm were introduced in the seminal work due to Auer et al. [10].

Combinatorial multi-armed bandit is a generalization of the standard multi-armed bandit model in which a subset of the base arms is selected rather than a single arm, where a feasible subset is called a super arm and it has to be subject to some combinatorial constraints.

There is a series of work on the stochastic combinatorial multi-armed bandit. By generalizing the idea of UCB method, Gai et al. [11] devised the Combinatorial UCB algorithm, denoted as CUCB for short here. Chen et al. [12] showed that the CUCB algorithm has an $O(\frac{1}{\Delta}m^2 d \log T)$ instance dependent regret bound, where d is the number of base arms, m is the maximal size of the super arms, and Δ is the expected loss/reward gap between the optimal and the best suboptimal solutions. Kveton et al. [13] further improved this regret bound to $O(\frac{1}{\Delta}md \log T)$, which matched the lower bound, and provided an instance independent regret bound as $O(\sqrt{mdT \log T})$. Some especially interesting cases were also studied, for example, the combinatorial bandit with knapsack constraint [14] or with matroid constraint [15]. Of which, the case with matroid constraint was further improved to achieve an $O(\frac{1}{\Delta}d \log T)$ regret bound [15].

Adversarial combinatorial bandit is often discussed with the case of the routing problem, including the work due to Takimoto and Warmuth [16] for the full information setting, Awerbuch and Kleinberg [17] and McMahan and Blum [18] for the full bandit setting. There is a series of work discussing the general full bandit cases [2,19–23]. Although in principle the full bandit algorithm can solve the semi-bandit problems and still achieve an $O(T^{\frac{1}{2}})$ regret bound, their regret is worse in terms of other factors of the problem and further their computation is inefficient for NP-hard problems.

The first work contributing to the semi-bandit setting gives credit to György et al. [1], in which the typical EWA algorithm [24–26] was adopted and it achieved an $O(md\sqrt{dT})$ high probability regret bound. However, the EWA method is proved to be suboptimal and the promising approach that provides the best regret guarantee of $O(\sqrt{mdT})$ is the OSMD algorithm [2], which also matches the lower bound.

The FPL method discussed in this paper was first proposed by Hannan [27] in the context of game theory and reinvented by Kalai and Vempala [5] for the full information setting. Kalai and Vempala [5] also discussed the possibility of utilizing an approximation oracle to solve the full information problem but there are no specific results and analysis provided. The simplified version of our work could be seen as the complementation to their idea. As previously introduced, the FPL method was adopted by Neu and Bartók [3, 4] to the adversarial combinatorial semi-bandit problem and their method gave an $\widetilde{O}(m\sqrt{dT})$ regret bound. Although this bound does not match the result of the OSMD approach, the FPL method has its practical value since it is quite simple and computationally efficient by leveraging the existing work on combinatorial optimization as oracles.

2 Problem Statement and Preliminaries

In this paper, we denote a super arm (i.e. a subset of base arms) or an action with a d-dimensional binary vector, 1 for chosen base arms and 0 for non-chosen base arms. The set of all feasible super arms (feasible actions/decisions), which is a subset of the power set of base arms, is mathematically a subset of all d-dimensional binary vectors, denoted as $\mathcal{S} \subseteq \{0,1\}^d$. In addition, we assume $\|\boldsymbol{v}\|_1 \leqslant m$ for all $\boldsymbol{v} \in \mathcal{S}$, that is, a feasible super arm consists of at most m base arms.

An adversarial combinatorial semi-bandit problem could be viewed as a repeated game in T rounds played between a player and an adversary as follows. In each round $t = 1, 2, \ldots, T$, the player chooses an action $\boldsymbol{V}_t \in \mathcal{S}$. Simultaneously, the adversary chooses a d-dimensional loss vector $\boldsymbol{\ell}_t \in [0,1]^d$, which may depend on $\boldsymbol{V}_1, \ldots, \boldsymbol{V}_{t-1}$. Then the player incurs a loss $\boldsymbol{V}_t^{\mathsf{T}}\boldsymbol{\ell}_t$. In addition to the loss, the player also observes the value of loss $\ell_{t,i}$ if and only if $V_{t,i} = 1$. More succinctly, the player observes a vector $\boldsymbol{V}_t \circ \boldsymbol{\ell}_t$ as the feedback information, where $\circ$ is the element-wise multiplication. The player's goal is to minimize its total loss over T rounds.

Following the standard practice in online learning, the performance of the player's algorithm is measured in terms of the *regret*, defined as

$$R_T = \mathbb{E}\left[\sum_{t=1}^{T} \boldsymbol{V}_t^{\mathsf{T}}\boldsymbol{\ell}_t\right] - \min_{\boldsymbol{v}\in\mathcal{S}} \mathbb{E}\left[\sum_{t=1}^{T} \boldsymbol{v}^{\mathsf{T}}\boldsymbol{\ell}_t\right],$$

where the expectation is with respect to the randomness of the

algorithm since it is well known that a randomized algorithm is necessary and a deterministic algorithm cannot guarantee a sublinear regret in the adversarial settings. Intuitively, the regret is the gap in the expected total loss between the learning algorithm and the best fixed action in hindsight. A good algorithm is required to have a sublinear regret, that is, the average regret per round tends to zero as the round number T tends to infinity.

When the offline minimization problem with a known loss vector is NP-hard and only an approximation algorithm is available, we do not expect a sublinear regret for the online problem since the approximation algorithm does not guarantee the optimal offline solution in each round. In this case, we could relax the regret definition by only comparing against a factor γ ($\gamma \geqslant 1$) times the best solution in hindsight, leading to the following definition of the *γ-scaled regret*:

$$R_T^{\gamma} = \mathbb{E}\left[\sum_{t=1}^{T} \boldsymbol{V}_t^{\mathsf{T}} \boldsymbol{\ell}_t\right] - \gamma \min_{\boldsymbol{v}\in\mathcal{S}} \mathbb{E}\left[\sum_{t=1}^{T} \boldsymbol{v}^{\mathsf{T}} \boldsymbol{\ell}_t\right]. \tag{1}$$

While the above statement is abstract, we would like to provide a more intuitive understanding with an example of the vertex cover problem, which will be discussed in detail in Section 5 on experiments.

In a traffic network, every day the department monitors the traffic condition of each road by allocating monitors in some nodes. A road is monitored if at least one of its nodes is allocated with a monitor. The feasible action set $\mathcal{S}$ is the class of subsets of nodes satisfying that all roads could be monitored if monitors are allocated in such a subset of nodes. An action $\boldsymbol{v} \in \mathcal{S}$ is one of such feasible subsets, with binary representation. There is some cost by allocating a monitor in a node and the cost may change over time. If we present the cost in each node on day t as a vector $\boldsymbol{\ell}_t$, this is the loss vector we discuss above. Obviously, if an action $\boldsymbol{V}_t$ is taken on day t, the loss the department suffers on this day is $\boldsymbol{V}_t^{\mathsf{T}}\boldsymbol{\ell}_t$.

The stochastic model assumes that the loss vector $\boldsymbol{\ell}_t$ is an i.i.d. sample from a stationary distribution. However, this assumption does not always hold in practice. In this case, we simply make no assumption on the generation of the loss vector, leading to the adversarial setting we discuss here.

Regarding the feedback information, it may be not available to collect the cost information if a node is not selected, leading to the bandit or semi-bandit feedback settings. Obviously the semi-bandit model is more practical in this case than the full bandit model since the cost of each node should be available every day rather than just a single value as the total loss.

At last in this section, we turn to briefly introduce the theory of approximation algorithms [28,29], which plays an important role in our work.

Many combinatorial optimization problems are NP-hard and they are not expected to have a polynomial-time algorithm under the widely believed conjecture that P $\neq$ NP. It is sensible to sacrifice the performance for the computational efficiency and look for a near-optimal solution with polynomial time. In the language of minimization problems, if an algorithm running in polynomial time can guarantee a solution that is not greater than a factor α times the optimal solution in the objective, it is called an approximation algorithm with the approximation ratio α.

Some NP-hard problems allow being approximated to any required degree. For such problems, if an algorithm can guarantee an approximation ratio of $1+\varepsilon$ for any fixed $\varepsilon > 0$ and its running time is polynomial to the problem size and $1/\varepsilon$, it is said to be a *fully polynomial time approximation scheme (FPTAS)*. Some important problems such as the knapsack problem admit an FPTAS.

Notation remark In this paper, a vector is represented with boldface, e.g. $\boldsymbol{V}_t$, and its i-th component is denoted with a subscript i in normal mathematical font, e.g. $V_{t,i}$. Another thing should pay attention to is to distinguish T and T. The italic T denotes the total number of rounds and T always appears in superscripts, denoting the matrix or vector transpose.

3 Our Algorithm

The key idea of our approach is to utilize an approximation algorithm rather than an inefficient exact optimization oracle in the FPL method. More specifically, suppose $\widehat{\boldsymbol{\ell}_t}$ is an estimation of $\boldsymbol{\ell}_t$ ($t = 1, \ldots, T$) that will be introduced later, and $O_A^{\alpha}(\cdot)$ is an α-approximation oracle that takes a loss vector $\boldsymbol{L} \in \mathbb{R}^d$ as input and outputs an approximation solution $O_A^{\alpha}(\boldsymbol{L}) \in \mathcal{S}$ satisfying

$$O_A^{\alpha}(\boldsymbol{L})^{\mathsf{T}}\boldsymbol{L} \leqslant \alpha \min_{\boldsymbol{v}\in\mathcal{S}} \boldsymbol{v}^{\mathsf{T}}\boldsymbol{L}.$$

Our algorithm chooses the action in round t by

$$\boldsymbol{V}_t = O_A^{\alpha}\left(\widehat{\boldsymbol{L}}_{t-1} + \boldsymbol{Z}_t\right), \tag{2}$$

in which $\widehat{\boldsymbol{L}}_{t-1} = \sum_{s=1}^{t-1} \widehat{\boldsymbol{\ell}}_s$, and $\boldsymbol{Z}_t \in \mathbb{R}^d$ is a perturbation vector whose components are i.i.d. random variables following a uniform distribution over $[0, u]$ and u is a parameter to be determined.

Our algorithm is quite different from the previous work. First, we utilize an approximation algorithm as the offline oracle rather than an exact optimization algorithm, which is efficient for NP-hard problems. Second, we perturb the cumulative loss by adding a nonnegative uniformly random vector and it can ensure that our method always works well even though the oracle only allows a nonnegative loss vector, while Neu and Bartók's work of subtracting an exponential random vector cannot make such a guarantee.

Loss estimation is another key step in the algorithm. One challenge of solving the adversarial combinatorial semi-bandit problem is that $\boldsymbol{\ell}_t$ is not completely available. The common practice in bandit algorithms is to estimate $\boldsymbol{\ell}_t$ and a widely-used method is of the form $\widehat{\ell}_{t,i} = V_{t,i}\ell_{t,i}/q_{t,i}$, for $i = 1, 2, \ldots, d$. Here $q_{t,i} \triangleq \Pr[V_{t,i} = 1|\mathcal{F}_{t-1}]$ and $\mathcal{F}_{t-1}$ is the sigma-algebra induced by the history of interaction between the player and the adversary up to the end of round $t-1$. It is straightforward to verify that $\widehat{\ell}_{t,i}$ is an unbiased estimation of $\ell_{t,i}$.

However, a consequent obstacle is that $q_{t,i}$ is unavailable explicitly in the FPL method since $\boldsymbol{V}_t$ is produced by an oracle call. To tackle this problem, Neu and Bartók proposed an approach called the Geometric Resampling (GR) to estimate the reciprocal of the probability, namely $1/q_{t,i}$, which is recapped as follows.

Notice that $\widehat{\ell}_{t,i} = 0$ if $V_{t,i} = 0$ no matter what value $q_{t,i}$ is, so we focus on the case that $V_{t,i} = 1$. Since $q_{t,i}$ is the probability that $V_{t,i} = 1$ occurs, the term $1/q_{t,i}$ could be interpreted as the expectation of a geometric distribution. A random variable $K_{t,i}$ following this geometric distribution is the number of samples (coin flips) drawn according to (2) with i.i.d. copies of $\boldsymbol{Z}_t$ until the event $V_{t,i} = 1$ occurs again. Since $\mathbb{E}[K_{t,i}] = 1/q_{t,i}$, $K_{t,i}$ could be adopted as an estimation of $1/q_{t,i}$ and the estimated loss is computed as

$$\widehat{\ell}_{t,i} = K_{t,i}V_{t,i}\ell_{t,i}. \tag{3}$$

While we will show that the expected number of samples for evaluating $\boldsymbol{K}_t$ is bounded fairly well, the actual number of samples might be large and the worst-case running time is unbounded. To ensure that the sampling procedure will terminate within finite steps, the GR method introduces a remedy that cuts off the number of samples by an upper bound M. Although this treatment will introduce some bias to the estimation, it can be shown that by appropriately chosen M, it does not hurt too much.

Based on the above ideas, the complete algorithm combining the FPL method, the offline approximation oracle, and the GR technique is depicted in Algorithm 1.

Algorithm 1 FPL + GR + offline approximation oracle

Input: parameters u and M, approximation oracle $\mathcal{O}_A^\alpha(\cdot)$;
Initialization: $\widehat{\boldsymbol{L}}_0 = \boldsymbol{0} \in \mathbb{R}^d$;
for $t = 1, \ldots, T$ **do**
  Draw $\boldsymbol{Z}_t \in \mathbb{R}^d$ with i.i.d. components $Z_{t,i} \sim U(0, u)$;
  Choose the action by calling the approximation oracle

$$\boldsymbol{V}_t = \mathcal{O}_A^\alpha\left(\widehat{\boldsymbol{L}}_{t-1} + \boldsymbol{Z}_t\right);$$

  Play $\boldsymbol{V}_t$ and observe the semi-bandit feedback $\boldsymbol{V}_t \circ \boldsymbol{\ell}_t$;
  $\boldsymbol{K}_t = \boldsymbol{0} \in \mathbb{R}^d$, $\boldsymbol{h} = \boldsymbol{V}_t$;
  while $\|\boldsymbol{h}\|_\infty > 0$ and $\|\boldsymbol{K}_t\|_\infty < M$ **do**
    $\boldsymbol{K}_t = \boldsymbol{K}_t + \boldsymbol{h}$;
    Draw $\boldsymbol{Z}' \in \mathbb{R}^d$ with i.i.d. components $Z'_i \sim U(0, u)$;
    choose an auxiliary action $\boldsymbol{V}' = \mathcal{O}_A^\alpha\left(\widehat{\boldsymbol{L}}_{t-1} + \boldsymbol{Z}'\right)$;
    $\boldsymbol{h} = \boldsymbol{h} - \boldsymbol{h} \circ \boldsymbol{V}'$;
  end while
  $\widehat{\boldsymbol{\ell}}_t = \boldsymbol{K}_t \circ \boldsymbol{V}_t \circ \boldsymbol{\ell}_t$;
  $\widehat{\boldsymbol{L}}_t = \widehat{\boldsymbol{L}}_{t-1} + \widehat{\boldsymbol{\ell}}_t$;
end for

4 Theoretical Analysis

The following theorem states the performance guarantee of Algorithm 1. Since the optimal offline solution cannot be assured, we do not expect a sublinear regret but a reasonable γ-scaled regret where γ is related to the approximation ratio α of the offline oracle. Although the regret guarantee is unclear for general cases, we show that our approach at least works well for an important class of combinatorial optimization problems that admit an FPTAS.

Theorem 1. *Suppose an offline combinatorial optimization problem admits an FPTAS. For any specific* $0 < \varepsilon \leqslant 2$, $u > 0$, *and* $M > 0$, *Algorithm 1 for solving the corresponding adversarial combinatorial semi-bandit problem by calling the approximation oracle with* $\alpha = 1 + \frac{\varepsilon}{2T}$ *can guarantee the* $(1+\varepsilon)$*-scaled regret by*

$$R_T^{1+\varepsilon} \leqslant \frac{1}{2}(1+\varepsilon)mu + \frac{dmMT}{u} + \frac{dT}{eM}.$$

Particularly, by letting

$$u = \left(\frac{4d^2}{e(1+\varepsilon)^2 m}\right)^{\frac{1}{3}} T^{\frac{2}{3}} \quad \text{and} \quad M = \left(\frac{2d}{e^2(1+\varepsilon)m^2}\right)^{\frac{1}{3}} T^{\frac{1}{3}},$$

the $(1+\varepsilon)$*-scaled regret of Algorithm 1 is bounded as*

$$R_T^{1+\varepsilon} \leqslant 3\left(\frac{(1+\varepsilon)m^2d^2}{2e}\right)^{\frac{1}{3}} T^{\frac{2}{3}}.$$

This bound is based on the condition that the algorithm needs to know the time horizon T upfront. For the any-time setting, that is, T is not available to the algorithm before the game playing, we can use the standard doubling trick, merely leading to an additional constant factor to the bound.

As in usual bandit algorithms, our method can be seen as an exploration-exploitation schema. Seeing from Theorem 1, the perturbation variables have the order of $O(T^{\frac{2}{3}})$. Intuitively, in the first $O(T^{\frac{2}{3}})$ rounds, the random perturbation dominates and the algorithm takes more exploration. As the number of rounds increases, the cumulative loss vector takes over and the exploitation comes into play.

One may expect an $O(T^{\frac{1}{2}})$ regret bound as in the stochastic setting. However, we failed to achieve this and the obstacle is discussed in subsection 4.3.

Regarding the computational cost, Algorithm 1 is efficient and its running time is characterized in Theorem 2, which states that the expected number of calls to the oracle in each round is not greater than the size of the problem instance. Since the approximation oracle runs in polynomial time, the expected total running time is still polynomial to the whole problem size.

Theorem 2. *In Algorithm 1, the expected number of calls to the oracle in each round for the Geometric Resampling process is not greater than d. Formally,*

$$\mathbb{E}\left[\max_{1\leqslant i\leqslant d} K_{t,i}\right] \leqslant d.$$

Consequently, the expected total number of calls to the oracle is upper bounded by $(d+1)T$.

The running time will be much better in practice since the upper bound of the number of resampling M is usually relatively small than the problem size d. For example, for a moderate combinatorial optimization problem with $m = d = 100$, M is less than 15 even though the time horizon T is as large as 10^6, let alone that M is around 3 if $T = 10^4$.

4.1 Technical Lemmas

In this and the next subsections, we provide analysis to the above key results. We synthetically combine previous work and adopt techniques from [5], [30], [26], [3], and [4]. Although most of the tools are quite standard, there are indeed some challenges in the analysis for the semi-bandit setting with approximation oracles since some tricks suitable for exact optimization oracles no longer work well for approximation oracles. We make some key and necessary modification to the algorithm and analysis.

We organize the analysis as follows. For completeness and making this paper self-contained, we first list the technical lemmas and their proofs in this subsection. Some of them are standard tools and can be founded in the literature. Lemma 1 and Lemma 2 are from [4] and we rephrase them in our context. However, we make considerable modification to some existing results as Lemma 3 and Lemma 4 for adapting to the approximation oracles.

With these lemmas, we compose the main proof steps to Theorem 1 and Theorem 2 in subsection 4.2.

Lemma 1 characterizes the estimation bias introduced by the cutting upper bound M.

Lemma 1. *For all t and i, the Geometric Resampling method for loss estimation (3) satisfies*

$$\mathbb{E}\left[K_{t,i}|\mathcal{F}_{t-1}\right] = \frac{1-(1-q_{t,i})^M}{q_{t,i}},$$

and

$$\mathbb{E}\left[\widehat{\ell}_{t,i}|\mathcal{F}_{t-1}\right] = \left(1-(1-q_{t,i})^M\right)\ell_{t,i}.$$

Consequently for any fixed $\boldsymbol{v} \in \mathcal{S}$,

$$\mathbb{E}\left[\boldsymbol{v}^\mathsf{T}\widehat{\boldsymbol{\ell}}_t|\mathcal{F}_{t-1}\right] \leqslant \boldsymbol{v}^\mathsf{T}\boldsymbol{\ell}_t.$$

From this lemma, we can see that $K_{t,i}$ and $\widehat{\ell}_{t,i}$ are underestimations of $1/q_{t,i}$ and $\ell_{t,i}$ respectively. What interesting is that the upper bound of $\widehat{\ell}_{t,i}$ is as large as M while $\ell_{t,i}$ is upper bounded by 1. However, we can still use $K_{t,i}$ to replace $1/q_{t,i}$ and $\widehat{\ell}_{t,i}$ to replace $\ell_{t,i}$ since the estimation gaps decrease exponentially with respect to M and can be controlled.

Proof. The procedure of GR gives

$$\begin{aligned}\mathbb{E}[K_{t,i}|\mathcal{F}_{t-1}] &= \sum_{k=1}^{M-1} k(1-q_{t,i})^{k-1}q_{t,i} + \sum_{k=M}^{\infty} M(1-q_{t,i})^{k-1}q_{t,i} \\ &= \frac{1-(1-q_{t,i})^M}{q_{t,i}}.\end{aligned}$$

By the definition of $\widehat{\ell}_{t,i}$, we have

$$\begin{aligned}\mathbb{E}\left[\widehat{\ell}_{t,i}|\mathcal{F}_{t-1}\right] &= \mathbb{E}\left[K_{t,i}V_{t,i}\ell_{t,i}|\mathcal{F}_{t-1}\right] \\ &= \mathbb{E}\left[K_{t,i}|\mathcal{F}_{t-1}\right]\mathbb{E}\left[V_{t,i}|\mathcal{F}_{t-1}\right]\ell_{t,i} \\ &= \frac{1-(1-q_{t,i})^M}{q_{t,i}}\cdot q_{t,i}\ell_{t,i} \\ &= \left(1-(1-q_{t,i})^M\right)\ell_{t,i}.\end{aligned}$$

By the previous result that $\mathbb{E}\left[\widehat{\ell}_{t,i}|\mathcal{F}_{t-1}\right] \leqslant \ell_{t,i}$, we have

$$\mathbb{E}\left[\boldsymbol{v}^\mathsf{T}\widehat{\boldsymbol{\ell}}_t|\mathcal{F}_{t-1}\right] = \sum_{i=1}^{d} v_i\mathbb{E}\left[\widehat{\ell}_{t,i}|\mathcal{F}_{t-1}\right] \leqslant \sum_{i=1}^{d} v_i\ell_{t,i} = \boldsymbol{v}^\mathsf{T}\boldsymbol{\ell}_t.$$

□

For adopting tools for analyzing full information settings to semi-bandit settings, one trick is to replace the actual loss $\boldsymbol{\ell}_t$ with an estimation $\widehat{\boldsymbol{\ell}}_t$ as if it is the full information feedback. However, the term $\boldsymbol{V}_t^\mathsf{T}\widehat{\boldsymbol{\ell}}_t$ for replacing $\boldsymbol{V}_t^\mathsf{T}\boldsymbol{\ell}_t$ is difficult to handle since $\widehat{\boldsymbol{\ell}}_t$ depends on $\boldsymbol{V}_t$. A further trick to overcome this difficulty is to introduce an i.i.d. counterpart of $\boldsymbol{V}_t$, denoted as $\widetilde{\boldsymbol{V}}_{t-1}$, which has the identical mechanism as $\boldsymbol{V}_t$ with i.i.d. random perturbation and is independent to $\widehat{\boldsymbol{\ell}}_t$. Readers may see the main proof steps to Theorem 1 in subsection 4.2 for the detailed explanation to this trick and $\widetilde{\boldsymbol{V}}_{t-1}$. It is sufficient to understand the following lemma, which provides an upper bound of $\boldsymbol{V}_t^\mathsf{T}\boldsymbol{\ell}_t$ in terms of $\widetilde{\boldsymbol{V}}_{t-1}^\mathsf{T}\widehat{\boldsymbol{\ell}}_t$, with the assumption that $\widetilde{\boldsymbol{V}}_{t-1}$ is i.i.d. with $\boldsymbol{V}_t$ and independent to $\boldsymbol{\ell}_t$.

Lemma 2. *For all t,*

$$\mathbb{E}\left[\boldsymbol{V}_t^\mathsf{T}\boldsymbol{\ell}_t|\mathcal{F}_{t-1}\right] \leqslant \mathbb{E}\left[\widetilde{\boldsymbol{V}}_{t-1}^\mathsf{T}\widehat{\boldsymbol{\ell}}_t|\mathcal{F}_{t-1}\right] + \frac{d}{eM}.$$

Proof. Notice that $\widetilde{\boldsymbol{V}}_{t-1}$ is i.i.d. with $\boldsymbol{V}_t$ and independent with $\widehat{\boldsymbol{\ell}}_t$ in the condition of $\mathcal{F}_{t-1}$, we have

$$\begin{aligned}
\mathbb{E}\left[\widetilde{\boldsymbol{V}}_{t-1}^\mathsf{T}\widehat{\boldsymbol{\ell}}_t|\mathcal{F}_{t-1}\right] &= \sum_{i=1}^{d}\mathbb{E}\left[\widetilde{V}_{t-1,i}\widehat{\ell}_{t,i}|\mathcal{F}_{t-1}\right] \\
&= \sum_{i=1}^{d}\mathbb{E}\left[\widetilde{V}_{t-1,i}|\mathcal{F}_{t-1}\right]\mathbb{E}\left[\widehat{\ell}_{t,i}|\mathcal{F}_{t-1}\right] \\
&= \sum_{i=1}^{d} q_{t,i}\left(1-(1-q_{t,i})^M\right)\ell_{t,i} \\
&\geqslant \sum_{i=1}^{d} q_{t,i}\ell_{t,i} - \sum_{t=1}^{d} q_{t,i}(1-q_{t,i})^M \\
&\geqslant \mathbb{E}\left[\boldsymbol{V}_t^\mathsf{T}\boldsymbol{\ell}_t|\mathcal{F}_{t-1}\right] - \frac{d}{eM}.
\end{aligned}$$

The last inequality holds because $q_{t,i}(1-q_{t,i})^M \leqslant q_{t,i}e^{-q_{t,i}M} \leqslant \max_{q\in\mathbb{R}} qe^{-qM} = \frac{1}{eM}$ and the maximum is taken at $q = \frac{1}{M}$. □

The intuition of the following lemma is that the expectation of $\widetilde{\boldsymbol{V}}_{t-1}^\mathsf{T}\widehat{\boldsymbol{\ell}}_t$ is "close" to the expectation of $\widetilde{\boldsymbol{V}}_t^\mathsf{T}\widehat{\boldsymbol{\ell}}_t$ and their difference can be upper bounded by a controlled term. The statement and proof of Lemma 3 are quite different from the counterparts, Lemma 6 and Lemma 8 in [4] in which the perturbation follows the exponential distribution and the properties of the exact optimization oracle are highly utilized. Here we borrow some idea from [5] for the uniform distribution and adapt our analysis to the approximation oracle.

Lemma 3. *For all t,*

$$\mathbb{E}\left[\widetilde{\boldsymbol{V}}_{t-1}^\mathsf{T}\widehat{\boldsymbol{\ell}}_t|\mathcal{F}_{t-1}\right] \leqslant \mathbb{E}\left[\widetilde{\boldsymbol{V}}_t^\mathsf{T}\widehat{\boldsymbol{\ell}}_t|\mathcal{F}_{t-1}\right] + \frac{dmM}{u}.$$

Proof.

$$\begin{aligned}
\mathbb{E}\left[\widetilde{\boldsymbol{V}}_{t-1}^\mathsf{T}\widehat{\boldsymbol{\ell}}_t|\mathcal{F}_t\right] &= \int_{z\in[0,u]^d} O_A^\alpha\left(\widehat{\boldsymbol{L}}_{t-1}+z\right)^\mathsf{T}\widehat{\boldsymbol{\ell}}_t\cdot\frac{1}{u^d}\mathrm{d}z \\
&\leqslant \int_{z\in\widehat{\boldsymbol{\ell}}_t+[0,u]^d} O_A^\alpha\left(\widehat{\boldsymbol{L}}_{t-1}+z\right)^\mathsf{T}\widehat{\boldsymbol{\ell}}_t\cdot\frac{1}{u^d}\mathrm{d}z \\
&\quad + \int_{z\in[0,u]^d\setminus\left(\widehat{\boldsymbol{\ell}}_t+[0,u]^d\right)} O_A^\alpha\left(\widehat{\boldsymbol{L}}_{t-1}+z\right)^\mathsf{T}\widehat{\boldsymbol{\ell}}_t\cdot\frac{1}{u^d}\mathrm{d}z. \quad (4)
\end{aligned}$$

The first term in the right-hand side is actually $\mathbb{E}\left[\widetilde{\boldsymbol{V}}_t^\mathsf{T}\widehat{\boldsymbol{\ell}}_t|\mathcal{F}_t\right]$ since

$$\begin{aligned}
&\int_{z\in\widehat{\boldsymbol{\ell}}_t+[0,u]^d} O_A^\alpha\left(\widehat{\boldsymbol{L}}_{t-1}+z\right)^\mathsf{T}\widehat{\boldsymbol{\ell}}_t\cdot\frac{1}{u^d}\mathrm{d}z \\
=&\int_{z\in\widehat{\boldsymbol{\ell}}_t+[0,u]^d} O_A^\alpha\left(\widehat{\boldsymbol{L}}_t+z-\widehat{\boldsymbol{\ell}}_t\right)^\mathsf{T}\widehat{\boldsymbol{\ell}}_t\cdot\frac{1}{u^d}\mathrm{d}z \\
=&\int_{\boldsymbol{y}\in[0,u]^d} O_A^\alpha\left(\widehat{\boldsymbol{L}}_t+\boldsymbol{y}\right)^\mathsf{T}\widehat{\boldsymbol{\ell}}_t\cdot\frac{1}{u^d}\mathrm{d}\boldsymbol{y} \\
=&\mathbb{E}\left[\widetilde{\boldsymbol{V}}_t^\mathsf{T}\widehat{\boldsymbol{\ell}}_t|\mathcal{F}_t\right].
\end{aligned}$$

We turn to bound the second term of (4).

$$\int_{z\in[0,u]^d\setminus\left(\widehat{\boldsymbol{\ell}}_t+[0,u]^d\right)} O_A^\alpha\left(\widehat{\boldsymbol{L}}_{t-1}+z\right)^\mathsf{T}\widehat{\boldsymbol{\ell}}_t\cdot\frac{1}{u^d}\mathrm{d}z \quad (5)$$

$$\leqslant mM\int_{z\in[0,u]^d\setminus\left(\widehat{\boldsymbol{\ell}}_t+[0,u]^d\right)}\frac{1}{u^d}\mathrm{d}z \quad (6)$$

$$\leqslant mM\sum_{i=1}^{d}\int_{z_i\in\left[0,\widehat{\ell}_{t,i}\right],z_j\in[0,u],j\neq i}\frac{1}{u^d}\mathrm{d}z \quad (7)$$

$$=\frac{mM}{u}\sum_{i=1}^{d}\widehat{\ell}_{t,i},$$

where (6) holds since $\|\boldsymbol{v}\|_1 \leqslant m$ for all $\boldsymbol{v}\in\mathcal{S}$ and $\|\widehat{\boldsymbol{\ell}}_t\|_\infty \leqslant M$, and (7) is actually the union bound.

Combining the results above, we have

$$\mathbb{E}\left[\widetilde{\boldsymbol{V}}_{t-1}^\mathsf{T}\widehat{\boldsymbol{\ell}}_t|\mathcal{F}_t\right] \leqslant \mathbb{E}\left[\widetilde{\boldsymbol{V}}_t^\mathsf{T}\widehat{\boldsymbol{\ell}}_t|\mathcal{F}_t\right] + \frac{mM}{u}\sum_{i=1}^{d}\widehat{\ell}_{t,i}.$$

Taking expectation to be conditioned on $\mathcal{F}_{t-1}$ and by Lemma 1, we finally have

$$\mathbb{E}\left[\widetilde{\boldsymbol{V}}_{t-1}^\mathsf{T}\widehat{\boldsymbol{\ell}}_t|\mathcal{F}_{t-1}\right] \leqslant \mathbb{E}\left[\widetilde{\boldsymbol{V}}_t^\mathsf{T}\widehat{\boldsymbol{\ell}}_t|\mathcal{F}_{t-1}\right] + \frac{dmM}{u}.$$

□

Lemma 4 is a generalization of the standard tool "be-the-leader" lemma (Lemma 3.1 in [26]) with a linear loss to the cases that utilize an approximation oracle. A similar result is also mentioned in [5] but there is no analysis provided. Please note that a negative loss will fail the induction in the proof. This is another reason that we have to perturb the loss by adding nonnegative noise.

Lemma 4. *Assume $\mathcal{S}' \subseteq [0,\infty)^d$, $\boldsymbol{\ell}'_1, \ldots, \boldsymbol{\ell}'_T$ is an arbitrary vector sequence in $[0,\infty)^d$, and $\boldsymbol{V}'_t$ for $t = 1, \ldots, T$ satisfies*

$$\boldsymbol{V}_t'^{\mathsf{T}} \sum_{s=1}^{t} \boldsymbol{\ell}'_s \leqslant \alpha \min_{\boldsymbol{u}\in\mathcal{S}'} \boldsymbol{u}^{\mathsf{T}} \sum_{s=1}^{t} \boldsymbol{\ell}'_s,$$

for a constant $\alpha \geqslant 1$ and all $t = 1, \ldots, T$, then

$$\sum_{t=1}^{T} \boldsymbol{V}_t'^{\mathsf{T}} \boldsymbol{\ell}'_t \leqslant \alpha^T \min_{\boldsymbol{u}\in\mathcal{S}'} \boldsymbol{u}^{\mathsf{T}} \sum_{t=1}^{T} \boldsymbol{\ell}'_t.$$

Proof. First we would like to prove

$$\sum_{t=1}^{T} \boldsymbol{V}_t'^{\mathsf{T}} \boldsymbol{\ell}'_t \leqslant \alpha^{T-1} \boldsymbol{V}_T'^{\mathsf{T}} \sum_{t=1}^{T} \boldsymbol{\ell}'_t \tag{8}$$

with induction.

The case of $T = 1$ is trivial since $\boldsymbol{V}_1'^{\mathsf{T}} \boldsymbol{\ell}'_t \leqslant \alpha^0 \boldsymbol{V}_1'^{\mathsf{T}} \boldsymbol{\ell}'_t$. Assume the conclusion holds for $T-1$, that is,

$$\sum_{t=1}^{T-1} \boldsymbol{V}_t'^{\mathsf{T}} \boldsymbol{\ell}'_t \leqslant \alpha^{T-2} \boldsymbol{V}_{T-1}'^{\mathsf{T}} \sum_{t=1}^{T-1} \boldsymbol{\ell}'_t.$$

Because

$$\boldsymbol{V}_{T-1}'^{\mathsf{T}} \sum_{t=1}^{T-1} \boldsymbol{\ell}'_t \leqslant \alpha \min_{\boldsymbol{u}\in\mathcal{S}'} \boldsymbol{u}^{\mathsf{T}} \sum_{t=1}^{T-1} \boldsymbol{\ell}'_t \leqslant \alpha \boldsymbol{V}_T'^{\mathsf{T}} \sum_{t=1}^{T-1} \boldsymbol{\ell}'_t,$$

we have

$$\sum_{t=1}^{T-1} \boldsymbol{V}_t'^{\mathsf{T}} \boldsymbol{\ell}'_t \leqslant \alpha^{T-2} \cdot \alpha \boldsymbol{V}_T'^{\mathsf{T}} \sum_{t=1}^{T-1} \boldsymbol{\ell}'_t \leqslant \alpha^{T-1} \boldsymbol{V}_T'^{\mathsf{T}} \sum_{t=1}^{T-1} \boldsymbol{\ell}'_t.$$

Adding $\boldsymbol{V}_T'^{\mathsf{T}} \boldsymbol{\ell}'_T$ into both sides and with condition that $\alpha \geqslant 1$, we obtain

$$\sum_{t=1}^{T} \boldsymbol{V}_t'^{\mathsf{T}} \boldsymbol{\ell}'_t \leqslant \alpha^{T-1} \boldsymbol{V}_T'^{\mathsf{T}} \sum_{t=1}^{T-1} \boldsymbol{\ell}'_t + \boldsymbol{V}_T'^{\mathsf{T}} \boldsymbol{\ell}'_T \leqslant \alpha^{T-1} \boldsymbol{V}_T'^{\mathsf{T}} \sum_{t=1}^{T} \boldsymbol{\ell}'_t,$$

as stated in inequality (8).

Finally, because

$$\boldsymbol{V}_T'^{\mathsf{T}} \sum_{t=1}^{T} \boldsymbol{\ell}'_t \leqslant \alpha \min_{\boldsymbol{u}\in\mathcal{S}'} \boldsymbol{u}^{\mathsf{T}} \sum_{t=1}^{T} \boldsymbol{\ell}'_t,$$

we get the conclusion that

$$\sum_{t=1}^{T} \boldsymbol{V}_t'^{\mathsf{T}} \boldsymbol{\ell}'_t \leqslant \alpha^T \min_{\boldsymbol{u}\in\mathcal{S}'} \boldsymbol{u}^{\mathsf{T}} \sum_{t=1}^{T} \boldsymbol{\ell}'_t.$$

□

4.2 Proofs of the Main Results

We first analyze the performance guarantee by proving Theorem 1.

Proof of Theorem 1. Rather than directly analyzing the regret defined in equation (1), we study the bound of the expected loss gap between the algorithm and an arbitrary fixed action $\boldsymbol{v} \in \mathcal{S}$, denoted as

$$R_T^{1+\varepsilon}(\boldsymbol{v}) = \mathbb{E}\left[\sum_{t=1}^{T} \boldsymbol{V}_t^{\mathsf{T}} \boldsymbol{\ell}_t\right] - (1+\varepsilon)\mathbb{E}\left[\sum_{t=1}^{T} \boldsymbol{v}^{\mathsf{T}} \boldsymbol{\ell}_t\right].$$

It is obvious to see that a bound of $R_T^{1+\varepsilon}(\boldsymbol{v})$ is also a bound of $R_T^{1+\varepsilon}$ since $\boldsymbol{v}$ is chosen arbitrarily.

Since $\boldsymbol{\ell}_t$ is unknown to the algorithm, we would like to replace it with $\widehat{\boldsymbol{\ell}}_t$ as if it is the full information feedback to the player so that some tools of analyzing the full information setting could be utilized. Before that, it is helpful to decouple $\boldsymbol{V}_t$ and $\widehat{\boldsymbol{\ell}}_t$ as they are dependent. To this end, we introduce an on-looking virtual player that can peek one more step than the actual player and plays with an i.i.d. perturbation. More specifically, in round t, the virtual player chooses its action by

$$\widetilde{\boldsymbol{V}}_t = O_A^{\alpha}\left(\widehat{\boldsymbol{L}}_t + \widetilde{\boldsymbol{Z}}\right),$$

where $\widetilde{\boldsymbol{Z}}$ is an i.i.d. perturbation as $\boldsymbol{Z}_1$. It is easy to see that $\widetilde{\boldsymbol{V}}_{t-1}$ is i.i.d. with $\boldsymbol{V}_t$, thus $\widetilde{q}_{t-1,i} \triangleq \mathbb{E}\left[\widetilde{V}_{t-1,i}|\mathcal{F}_{t-1}\right] = \mathbb{E}\left[V_{t,i}|\mathcal{F}_{t-1}\right] = q_{t,i}$. Note that the virtual player is only introduced for the sake of analysis. The virtual actions $\widetilde{\boldsymbol{V}}_t$ will not be played and do not affect the actual process.

According to Lemma 1 and Lemma 2, $R_T^{1+\varepsilon}(\boldsymbol{v})$ is upper bounded in terms of $\widehat{\boldsymbol{\ell}}_t$ as

$$R_T^{1+\varepsilon}(\boldsymbol{v}) \leqslant \sum_{t=1}^{T} \mathbb{E}\left[\widetilde{\boldsymbol{V}}_{t-1}^{\mathsf{T}} \widehat{\boldsymbol{\ell}}_t\right] - (1+\varepsilon) \sum_{t=1}^{T} \mathbb{E}\left[\boldsymbol{v}^{\mathsf{T}} \widehat{\boldsymbol{\ell}}_t\right] + \frac{dT}{eM}.$$

By Lemma 3, we further have

$$R_T^{1+\varepsilon}(\boldsymbol{v}) \leqslant \sum_{t=1}^{T} \mathbb{E}\left[\widetilde{\boldsymbol{V}}_t^{\mathsf{T}} \widehat{\boldsymbol{\ell}}_t\right] - (1+\varepsilon) \sum_{t=1}^{T} \mathbb{E}\left[\boldsymbol{v}^{\mathsf{T}} \widehat{\boldsymbol{\ell}}_t\right] + \frac{dmMT}{u} + \frac{dT}{eM}.$$

We would like to use Lemma 4 to bound the first two terms. Let $\boldsymbol{\ell}'_1 = \widehat{\boldsymbol{\ell}}_1 + \widetilde{\boldsymbol{Z}}$, $\boldsymbol{\ell}'_2 = \widehat{\boldsymbol{\ell}}_2$, …, $\boldsymbol{\ell}'_T = \widehat{\boldsymbol{\ell}}_T$, and $\boldsymbol{V}'_t = \widetilde{\boldsymbol{V}}_t$. It is straightforward to verify that the conditions of Lemma 4 are satisfied and we get

$$\sum_{t=1}^{T} \widetilde{\boldsymbol{V}}_t^{\mathsf{T}} \widehat{\boldsymbol{\ell}}_t + \widetilde{\boldsymbol{V}}_1^{\mathsf{T}} \widetilde{\boldsymbol{Z}} \leqslant \alpha^T \sum_{t=1}^{T} \boldsymbol{v}^{\mathsf{T}} \widehat{\boldsymbol{\ell}}_t + \alpha^T \boldsymbol{v}^{\mathsf{T}} \widetilde{\boldsymbol{Z}}.$$

Reordering and taking expectation gives

$$\sum_{t=1}^{T}\mathbb{E}\left[\widetilde{\boldsymbol{V}}_t^{\top}\widehat{\boldsymbol{\ell}}_t\right]-(1+\varepsilon)\sum_{t=1}^{T}\mathbb{E}\left[\boldsymbol{v}^{\top}\widehat{\boldsymbol{\ell}}_t\right]\leqslant\mathbb{E}\left[\sum_{t=1}^{T}\widetilde{\boldsymbol{V}}_t^{\top}\widehat{\boldsymbol{\ell}}_t-\alpha^T\sum_{t=1}^{T}\boldsymbol{v}^{\top}\widehat{\boldsymbol{\ell}}_t\right] \tag{9}$$

$$\leqslant\alpha^T\mathbb{E}\left[\boldsymbol{v}^{\top}\widetilde{\boldsymbol{Z}}\right]$$

$$\leqslant\frac{1}{2}(1+\varepsilon)mu. \tag{10}$$

In which (9) and (10) hold since $\alpha^T=(1+\frac{\varepsilon}{2T})^T\leqslant e^{\frac{\varepsilon}{2T}\cdot T}=e^{\frac{\varepsilon}{2}}\leqslant 1+2\cdot\frac{\varepsilon}{2}=1+\varepsilon$ and this is further due to the facts that $e^x\geqslant 1+x$ for all $x\in\mathbb{R}$ and $e^x\leqslant 1+2x$ for $x\in[0,1]$. This is why we require $\varepsilon\leqslant 2$ in the theorem statement. Inequality (10) additionally holds by $\|\boldsymbol{v}\|_1\leqslant m$ for all $\boldsymbol{v}\in\mathcal{S}$ and $\mathbb{E}\left[\widetilde{Z}_i\right]=u/2$ for all $i=1,\ldots,d$.

Combining the results above and considering that $\boldsymbol{v}$ is chosen arbitrarily, we have the $(1+\varepsilon)$-scaled regret bound as

$$R_T^{1+\varepsilon}\leqslant\frac{1}{2}(1+\varepsilon)mu+\frac{dmMT}{u}+\frac{dT}{eM}. \tag{11}$$

By inequality of arithmetic and geometric means, we have

$$\begin{aligned}&\frac{1}{2}(1+\varepsilon)mu+\frac{dmMT}{u}+\frac{dT}{eM}\\ \geqslant&3\left(\frac{1}{2}(1+\varepsilon)mu\cdot\frac{dmMT}{u}\cdot\frac{dT}{eM}\right)^{\frac{1}{3}}\\ =&3\left(\frac{(1+\varepsilon)m^2d^2}{2e}\right)^{\frac{1}{3}}T^{\frac{2}{3}},\end{aligned}$$

where the equality holds when

$$\frac{1}{2}(1+\varepsilon)mu=\frac{dmMT}{u}=\frac{dT}{eM},$$

which implies,

$$u=\left(\frac{4d^2}{e(1+\varepsilon)^2m}\right)^{\frac{1}{3}}T^{\frac{2}{3}}\quad\text{and}\quad M=\left(\frac{2d}{e^2(1+\varepsilon)m^2}\right)^{\frac{1}{3}}T^{\frac{1}{3}}.$$

In conclusion, by choosing appropriate parameters u and M as above, we have a sublinear $(1+\varepsilon)$-scaled regret of order $O(T^{\frac{2}{3}})$ for any specific $\varepsilon\in(0,2]$ as

$$R_T^{1+\varepsilon}\leqslant 3\left(\frac{(1+\varepsilon)m^2d^2}{2e}\right)^{\frac{1}{3}}T^{\frac{2}{3}}.$$

□

And then we turn to the running time analysis by proving Theorem 2.

Proof of Theorem 2. Notice that $\max_{1\leqslant i\leqslant d}K_{t,i}$ is the number of samples for the GR process in round t and its conditional expectation with respect to a fixed $\boldsymbol{V}_t=\boldsymbol{v}$ is

$$\begin{aligned}\mathbb{E}\left[\max_{1\leqslant i\leqslant d}K_{t,i}|\mathcal{F}_{t-1},\boldsymbol{V}_t=\boldsymbol{v}\right]&=\mathbb{E}\left[\max_{1\leqslant i\leqslant d}K_{t,i}v_i|\mathcal{F}_{t-1},\boldsymbol{V}_t=\boldsymbol{v}\right]\\ &\leqslant\mathbb{E}\left[\sum_{i=1}^{d}K_{t,i}v_i|\mathcal{F}_{t-1},\boldsymbol{V}_t=\boldsymbol{v}\right]\\ &=\sum_{i=1}^{d}v_i\mathbb{E}\left[K_{t,i}|\mathcal{F}_{t-1}\right]\\ &\leqslant\sum_{i=1}^{d}\frac{v_i}{q_{t,i}}.\end{aligned}$$

Therefore,

$$\begin{aligned}\mathbb{E}\left[\max_{1\leqslant i\leqslant d}K_{t,i}|\mathcal{F}_{t-1}\right]&=\sum_{\boldsymbol{v}\in\mathcal{S}}\Pr\left[\boldsymbol{V}_t=\boldsymbol{v}|\mathcal{F}_{t-1}\right]\mathbb{E}\left[\max_{1\leqslant i\leqslant d}K_{t,i}|\mathcal{F}_{t-1},\boldsymbol{V}_t=\boldsymbol{v}\right]\\ &\leqslant\sum_{\boldsymbol{v}\in\mathcal{S}}\Pr\left[\boldsymbol{V}_t=\boldsymbol{v}|\mathcal{F}_{t-1}\right]\sum_{i=1}^{d}\frac{v_i}{q_{t,i}}\\ &=\sum_{i=1}^{d}\frac{1}{q_{t,i}}\sum_{\boldsymbol{v}\in\mathcal{S}}\Pr\left[\boldsymbol{V}_t=\boldsymbol{v}|\mathcal{F}_{t-1}\right]v_i\\ &=\sum_{i=1}^{d}\frac{1}{q_{t,i}}\cdot q_{t,i}\\ &=d.\end{aligned}$$

Since there is one additional oracle calling for choosing the actual action in each round and there are totally T rounds, we simply get the upper bound of the expected total number of calls to the oracle as $(d+1)T$. □

4.3 Discussions

In stochastic combinatorial semi-bandit, replacing the exact oracle in the CUCB algorithm with an approximation oracle keeps the order of regret as long as the regret is defined by a baseline scaled with the corresponding approximation ratio, i.e., the scaled regret [12, 14, 15].

However, we failed to achieve a similar result in the adversarial setting since there is some obstacle to this goal in our analysis. The cause of the $O(T^{\frac{2}{3}})$ scaled regret bound in our result is that we have to tune three terms with two hyperparameters u and M, see the statement in Theorem 1 and the inequality (11) in its proof. One key solution is to remove the term M in Lemma 3. Neu and Bartók's work employs some sophisticated tricks that highly depend on the properties of the exact oracle to achieve a better result similar to our Lemma 3, see Lemma 8 in [4]. However, we cannot adapt these tricks to the case with an approximation oracle. On the other hand, if we do not upper bound $\widehat{\ell}_{t,i}$ with M and leave it in equation (6), we have to upper bound $\mathbb{E}[\widehat{\ell}_{t,i}^2]$. However,

while $\mathbb{E}[\widehat{\ell}_{t,i}]$ can be easily bounded, $\mathbb{E}[\widehat{\ell}^2_{t,i}]$ is hard to bound since $\mathbb{E}[\widehat{\ell}^2_{t,i}] = \mathbb{E}[K^2_{t,i}V^2_{t,i}\ell^2_{t,i}] = \mathbb{E}[K^2_{t,i}V_{t,i}\ell^2_{t,i}]$. The term $V_{t,i}$ can only help eliminate one $K_{t,i}$ and the remaining $K_{t,i}$ can be upper bounded by $\frac{1}{q_{t,i}}$, which is unbounded actually, or simply M again. Therefore, so far we have to upper bound one $\widehat{\ell}_{t,i}$ with M simply, leading to an $O(T^{\frac{2}{3}})$ regret bound. In this sense, a bounded M is necessary in our analysis while it is optional in Neu and Bartók's work.

5 Experiments

In addition to the theoretical guarantee of our algorithm in the specific case, we also conduct a series of experiments to demonstrate the performance of our algorithm and to obtain more insight into the adversarial combinatorial semi-bandit problem. Subsection 5.1 introduces the experimental settings. The experimental results and comparison are reported in subsection 5.2.

5.1 Experimental Settings

5.1.1 Problems

We consider two NP-hard combinatorial optimization problems, one with FPTAS and the other having a constant 2 approximation algorithm. The first is for being consistent with the theoretical setting and the later is for testing the performance of our algorithm for more general cases.

Although the knapsack problem is a typical case that admits an FPTAS, for being consistent with the minimization setting in the algorithm and theoretical parts, we consider the complementary problem to the knapsack problem. We simply call it the *shopping problem* since it can be described as the following scenario.

Shopping problem There are n items, each with a price p_i and a functional value v_i. Both are positive real numbers. For completing a task, one has to buy a subset of the items (each one can be selected at most once), such that the total functional value of the selected items is not less than a required lower bound V and their total price is as little as possible. Formally, it is the following optimization problem:

$$\begin{aligned} &\min_{x_1,\dots,x_n} \sum_{i=1}^{n} p_i x_i, \\ &\text{s.t.} \sum_{i=1}^{n} v_i x_i \geqslant V, \\ &\quad x_i \in \{0,1\}, i = 1,\dots,n. \end{aligned}$$

Like the knapsack problem, if p_i's or v_i's are integers, there is a pseudo-polynomial dynamic programming (DP) algorithm. Further, under a mild and reasonable assumption that the price of the most expensive item will not be much greater than that of the cheapest item, formally $\frac{\max_i p_i}{\min_j p_j} = O(\text{poly}(n))$, the shopping problem admits an FPTAS. For simplicity in our experiment, we assume that the price of the most expensive item is not 10 times greater than that of the cheapest item.

Since the input data are real numbers and the DP algorithm does not apply directly, the exact oracle takes a brute-force search strategy.

The online learning variant to the shopping problem is that the functional values v_i's and V are fixed, while the prices p_i's are designated by an adversary in each round.

The other problem we consider is the *vertex cover problem*, which is a standard case in the literature on approximation algorithms and it is described as follows.

Vertex cover problem $G = (V, E)$ is an undirected graph with n vertices. Each vertex $v_i \in V$ has a positive price p_i. We would like to select a subset of the vertices such that each edge is associated with at least one selected vertex and the total price of the selected vertices is as little as possible. It is formally the following combinatorial optimization problem:

$$\begin{aligned} &\min_{x_1,\dots,x_n} \sum_{i=1}^{n} p_i x_i, \\ &\text{s.t.}\ x_i + x_j \geqslant 1, (v_i, v_j) \in E, \\ &\quad x_i \in \{0,1\}, i = 1,\dots,n. \end{aligned}$$

The pricing algorithm based on the primal-dual schema of linear programming has an approximation ratio of 2, which could be found in standard textbooks on approximation algorithms [28, 29]. The running time of the pricing algorithm is $O(n^2)$, much faster than the brute-force enumeration of $O(2^n)$ time, which was implemented in our experiment as the exact oracle.

The online learning variant to the vertex cover problem is that the graph G is fixed, while the price p_i of each vertex is designated by an adversary in every round.

5.1.2 Competing Algorithms

We denote our algorithm as *FPL+approx*, short for "Follow-the-Perturbed-Leader with an approximation oracle". The baseline we would like to compare with is certainly Neu and Bartók's work [3, 4], denoted as *FPL+exact*, short for "Follow-the-Perturbed-Leader with an exact oracle".

The combinatorial upper confidence bound algorithm [11,

12], CUCB[1)] for short, is a well-studied method and has a good performance guarantee in the stochastic setting. Considering that the stochastic setting is a special case of more general adversarial settings, we also compare our work with the CUCB-like algorithms, including *CUCB+exact*, short for "CUCB with an exact oracle", and *CUCB+approx*, short for "CUCB with an approximation oracle".

In short, we implemented four algorithms in the experiment, including FPL+exact, FPL+approx, CUCB+exact, and CUCB+approx.

5.1.3 Adversarial Data Policies

It is a big challenge to evaluate an online learning algorithm without live data, particular for adversarial settings, since it is quite hard to design the worst adversary to fool an algorithm. In this work, we design three adversarial data policies to test the algorithms.

The first approach, denoted as *stochastic adversary*, is the usual stochastic setting in line with the assumption in the stochastic bandit study, in which the price vector in each round is an i.i.d. sample from a stationary distribution.

The second data policy, called *adversary against history*, is an adversarial data generation method that fools an algorithm according to its past behaviors. Specifically, let $X_{t-1,i} = \sum_{s=1}^{t-1} x_{s,i}$ be the number of times the algorithm selected item i in the past $t-1$ rounds and $X_{\max} = \max_{1 \leqslant i \leqslant n} X_{t-1,i}$. When generating the data for round t, the adversary assign a high price to item i with probability $q_i = \frac{X_{t-1,i}}{X_{\max}}$, or a small price with probability $1 - q_i$. Intuitively, under this data policy, if some items are selected frequently in the past, the algorithm will most likely suffer a high loss if it still prefers these items in the future.

The third adversarial data policy, called *adversary against the future*, fools an algorithm according to the prediction of its next behavior. In this method, we assume that the adversary knows the algorithm without the player's random numbers (if there is any). When generating the data for round t, the adversary will first simulate the player with its own random numbers (if it is necessary and they are i.i.d. to the player's), producing a virtual solution $\boldsymbol{x}'$. The adversary set a high price to item i if $x'_i = 1$, or a small price if $x'_i = 0$. Obviously, this adversary is extremely unfriendly to deterministic algorithms such as CUCB since they suffer a high loss every round.

[1)] We keep the term "upper confidence bound" for being consistent with previous work, although we actually compute lower confidence bounds for minimization problems.

Please note that for a specific adversarial data policy (except the stochastic one), the adversary takes the algorithm as input as part of its data generation process. So the price data to various algorithms may be quite different, leading to different behaviors of the algorithms.

5.1.4 Evaluation Criteria

We evaluate the algorithms in two aspects. The first is the computational efficiency in terms of per round running time and the second is to test their performance in terms of (exact) regret.

We do not compute the scaled regret by multiplying the offline optimal solution with the approximation factor for the following reasons. The first is for a fair comparison since the regret and the scaled regret have different baselines and it is not appropriate to compare them directly. Moreover, the baseline of multiplying an approximation factor to the offline optimal solution is quite weak and the algorithm with an approximation oracle usually outperforms it, leading to negative regrets. Last but not least, we would like to see how much additional price shall we pay for computational efficiency with an approximation oracle rather than an exact oracle.

What should be mentioned is that we do not employ the total loss as a measurement, which is often used in the stochastic setting. We argue that while the total loss is appropriate for the stochastic setting, it is not a suitable evaluation in the adversarial setting. In the stochastic case, the expected offline optimal solution is identical and irrelevant to the algorithm so that the total loss is consistent with the regret. However, in the adversarial setting, there may be no consistent optimal solution over different runs and algorithms, leading to inconsistency between the total loss and the regret.

5.1.5 Instance and Parameter Configuration

For the shopping problem, we ran an instance with 28 items for testing the running time of each algorithm and we ran a relatively small instance with 10 items for evaluating the regret since we could not suffer the extremely long running time from the brute-force exact oracle while we had to run tens of thousands of trials for every setting. For generating a problem instance, we assigned the functional value v_i to item i randomly from $(0, 1)$, and then we set V as about 50% of the total functional values of all items.

For the vertex cover problem, the instance for testing the running time has 64 vertices and we evaluated the regret with a graph of 20 vertices for a similar reason. For a given prob-

lem size n, we generated a connected random graph according to the Erdős-Rényi model with the probability $\frac{\ln n}{n}$.

We simply employed an approximation oracle with an approximation ratio of 101% for the FPL+approx and CUCB+approx algorithms to the shopping problem.

Regarding the adversarial price assignment, the high prices were all 1 in both problems. The low prices were set to be 0.1 for the shopping problem and they were 0 for the vertex cover problem.

In the experimental results, the data of regret are the average from 100 trials and the data of running time are the average from 10 trials.

5.2 Experimental Results

The experimental results are illustrated in Figure 1 to Figure 3.

We first analyze the computational cost. Figure 1 shows the average running time per round in log scale of different algorithms to the two problems with respect to the time horizon T. Obviously, the algorithms employing approximation oracles run much faster than that calling exact oracles. The difference in order of magnitudes is as large as 10^2 to 10^3 in the shopping problem and 10^5 to 10^6 in the vertex cover problem.

Careful readers may note that the per round running time of the FPL+exact method increases along the time horizon T. The reason is that the upper bound of the number of resampling M in the FPL algorithm increases with T. This makes the FPL+exact approach further inefficient. The similar reason explains why our FPL+approx algorithm runs a little slower than the CUCB+approx method, since the CUCB-like approaches only call the oracle once every round.

And then we turn to the regret. Seeing from Figure 2, if the problem admits an FPTAS, our FPL+approx method performs closely to the FPL+exact algorithm and it outperforms the CUCB-like algorithms in the non-stochastic settings.

Figure 3 shows that our FPL+approx algorithm still outperforms the CUCB+ approx method even though the problem does not admit an FPTAS, except in the stochastic setting. Our method even performs better than the CUCB+exact algorithm in the adversary-against-future data policy.

Our experimental results provide comprehensive and reasonable insight for algorithm selection for the combinatorial semi-bandit problem: (1) If the computational cost is not a bottleneck, the FPL+exact algorithm works best in the non-stochastic adversarial setting while the CUCB+exact method is an excellent choice in the stochastic setting; (2) If the computational cost matters, our FPL+approx applies in the non-stochastic adversarial setting and the CUCB+approx should be selected in the stochastic setting.

6 Conclusion and Future Work

Combinatorial optimization in environments with uncertainty is an interesting and important problem in theory and practice. In this paper we study a special case, the adversarial combinatorial semi-bandit problem, which is useful for many applications. Although various algorithms have been developed for this problem, solving this problem efficiently is still a challenge.

Following the previous work, we propose a variant of the FPL method that employs an approximation algorithm as an offline oracle and perturbs the collected data by adding nonnegative noise.

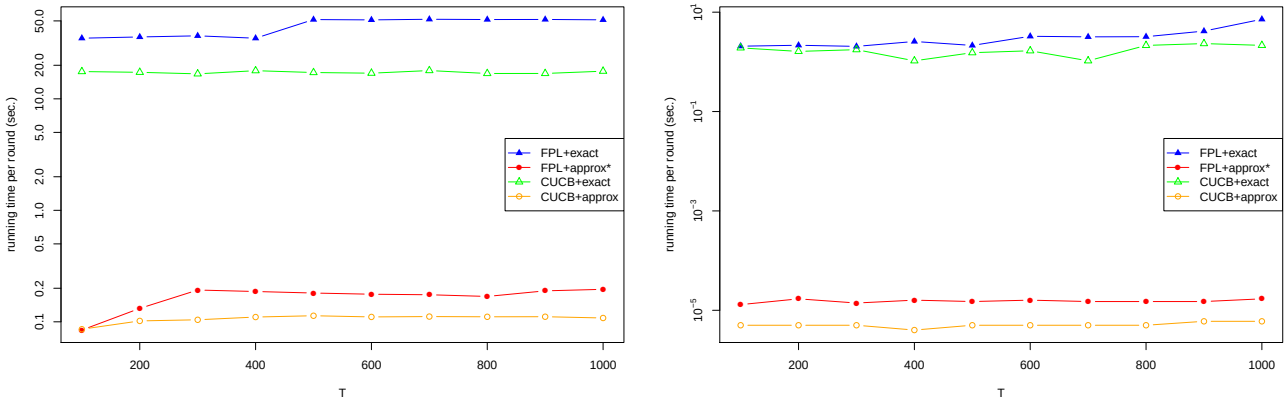
Our approach is simple and efficient. The expected number of calls to the oracle in each round is not greater than the offline problem size and the expected total running time is polynomial to the entire problem size.

Our algorithm can provide sublinear $(1+\varepsilon)$-scaled regret guarantee of order $O(T^{\frac{2}{3}})$ for any $\varepsilon \in (0, 2]$ for an important class of combinatorial optimization problems that admit an FPTAS. This result is significant since some important problems, such as the knapsack problem and some special scheduling problems, admit an FPTAS, and many applications can be reduced to such typical problems. The analysis is challenging since some techniques highly depending on the properties of exact oracles do not work for approximation oracles. We made some key and necessary modification to the analysis to adapt to the approximation oracle.

Our algorithm works well when the oracle only accept nonnegative inputs, while the original FPL method does not work in this case.

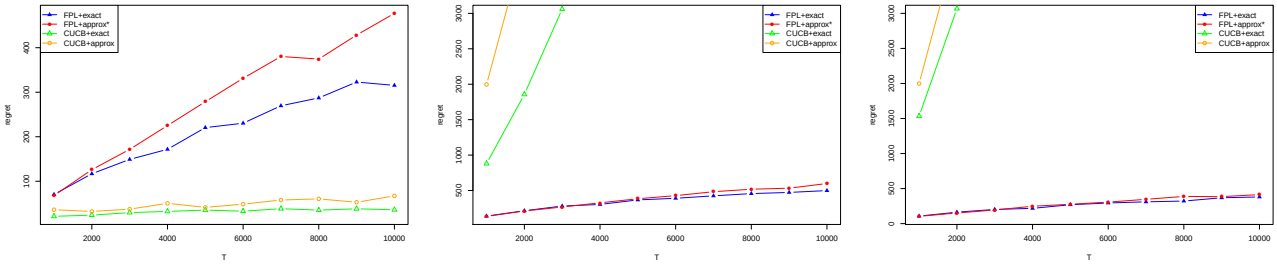
We conducted a series of experiments to demonstrate the performance of our algorithm. Seeing from the experimental results, our method runs much faster than that with an exact oracle. In terms of regret, our algorithm performs closely to the FPL method with an exact oracle if the problem admits an FPTAS, and it outperforms the CUCB-like algorithms in non-stochastic adversarial settings even in general cases.

Regarding the future work, improving the scaled regret bound to $O(T^{\frac{1}{2}})$ is considerable. In addition, analyzing our algorithm for general cases is still unsolved. We also notice that there are some recent tries to solve the online linear optimization problem with approximation oracles for the full



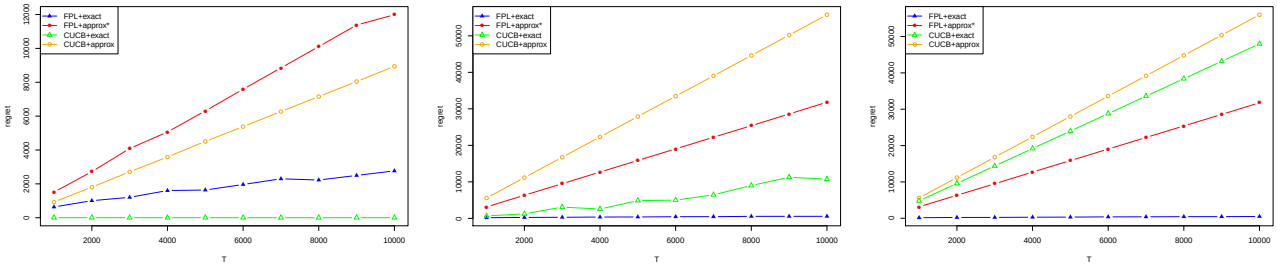
(a) the shopping problem (b) the vertex cover problem

Fig. 1: the running time per round (in log scale) of different algorithms



(a) stochastic adversary (b) adversary against history (c) adversary against future

Fig. 2: Regret to the shopping problem



(a) stochastic adversary (b) adversary against history (c) adversary against history

Fig. 3: Regret to the vertex cover problem

information setting, e.g. Kakade, Kalai, and Ligett [31], Garber [32], Hazan et al. [33]. Although these approaches can provide regret guarantee for general cases, but it seems that such methods are complicated and not so efficient. For example, the oracle complexity is related to the time horizon T. It is also very interesting to adopt these approaches to solve the adversarial combinatorial semi-bandit problem efficiently.

Acknowledgement

We would like to thank Weidong Ma from MSRA for the helpful discussion on approximation algorithm analysis.

This work was supported in part by the National Nat-

ural Science Foundation of China Grants No. 61832003, 61761136014, 61872334, the 973 Program of China Grant No. 2016YFB1000201, and K.C.Wong Education Foundation.

References

1. András György, Tamás Linder, Gábor Lugosi, and György Ottucsák. The on-line shortest path problem under partial monitoring. *Journal of Machine Learning Research*, 8(Oct):2369–2403, 2007.
2. Jean-Yves Audibert, Sébastien Bubeck, and Gábor Lugosi. Regret in online combinatorial optimization. *Mathematics of Operations Research*, 39(1):31–45, 2013.
3. Gergely Neu and Gábor Bartók. An efficient algorithm for learning with semi-bandit feedback. In *International Conference on Algorithmic Learning Theory*, pages 234–248. Springer, 2013.
4. Gergely Neu and Gábor Bartók. Importance weighting without importance weights: an efficient algorithm for combinatorial semi-bandits. *Journal of Machine Learning Research*, 17(154):1–21, 2016.
5. Adam Kalai and Santosh Vempala. Efficient algorithms for online decision problems. In *Learning Theory and Kernel Machines*, pages 26–40. Springer, 2003.
6. Wei Wang and Zhi-Hua Zhou. Crowdsourcing label quality: a theoretical analysis. *Science China Information Sciences*, 58(11):1–12, 2015.
7. William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933.
8. Herbert Robbins. Some aspects of the sequential design of experiments. In *Herbert Robbins Selected Papers*, pages 169–177. Springer, 1985.
9. Peter Auer, Nicolò Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002.
10. 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.
11. Y. Gai, B. Krishnamachari, and R. Jain. Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations. *IEEE/ACM Transactions on Networking*, 20(5):1466–1478, Oct 2012.
12. Wei Chen, Yajun Wang, Yang Yuan, and Qinshi Wang. Combinatorial multi-armed bandit and its extension to probabilistically triggered arms. *Journal of Machine Learning Research*, 17(50):1–33, 2016. A preliminary version appeared as Chen, Wang, and Yuan, "combinatorial multi-armed bandit: General framework, results and applications", ICML'2013.
13. Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvari. Tight regret bounds for stochastic combinatorial semi-bandits. In *AISTATS*, 2015.
14. Karthik Abinav Sankararaman and Aleksandrs Slivkins. Combinatorial semi-bandits with knapsacks. In *International Conference on Artificial Intelligence and Statistics*, pages 1760–1770, 2018.
15. Branislav Kveton, Zheng Wen, Azin Ashkan, Hoda Eydgahi, and Brian Eriksson. Matroid bandits: Fast combinatorial optimization with learning. *arXiv preprint arXiv:1403.5045*, 2014.
16. Eiji Takimoto and Manfred K Warmuth. Path kernels and multiplicative updates. *Journal of Machine Learning Research*, 4(Oct):773–818, 2003.
17. Baruch Awerbuch and Robert D Kleinberg. Adaptive routing with end-to-end feedback: Distributed learning and geometric approaches. In *Proceedings of the thirty-sixth annual ACM symposium on Theory of computing*, pages 45–53. ACM, 2004.
18. H Brendan McMahan and Avrim Blum. Online geometric optimization in the bandit setting against an adaptive adversary. In *International Conference on Computational Learning Theory*, pages 109–123. Springer, 2004.
19. Varsha Dani, Sham M Kakade, and Thomas P Hayes. The price of bandit information for online optimization. In *Advances in Neural Information Processing Systems*, pages 345–352, 2008.
20. Jacob D Abernethy, Elad Hazan, and Alexander Rakhlin. Competing in the dark: An efficient algorithm for bandit linear optimization. In *Conference on Learning Theory*, 2009.
21. Nicolo Cesa-Bianchi and Gábor Lugosi. Combinatorial bandits. *Journal of Computer and System Sciences*, 78(5):1404–1422, 2012.
22. Sébastien Bubeck, Nicolo Cesa-Bianchi, and Sham M Kakade. Towards minimax policies for online linear optimization with bandit feedback. In *Conference on Learning Theory*, pages 41–1, 2012.
23. Richard Combes, Mohammad Sadegh Talebi Mazraeh Shahi, Alexandre Proutiere, et al. Combinatorial bandits revisited. In *Advances in Neural Information Processing Systems*, pages 2116–2124, 2015.
24. Avrim Blum. On-line algorithms in machine learning. *Lecture Notes in Computer Science*, pages 306–325, 1998.
25. Dean P Foster and Rakesh V Vohra. Regret in the on-line decision problem. *Games and Economic Behavior*, 29:7–35, 1999.
26. Nicolo Cesa-Bianchi and Gábor Lugosi. *Prediction, learning, and games*. Cambridge university press, 2006.
27. James Hannan. Approximation to bayes risk in repeated play. *Contributions to the Theory of Games*, 3:97–139, 1957.
28. Vijay V Vazirani. *Approximation algorithms*. Springer, 2001.
29. David P Williamson and Advid B Shmoys. *The design of approximation algorithms*. Cambridge University Press, 2010.
30. Jan Poland. Fpl analysis for adaptive bandits. In *International Symposium on Stochastic Algorithms*, pages 58–69. Springer, 2005.
31. Sham M Kakade, Adam Tauman Kalai, and Katrina Ligett. Playing games with approximation algorithms. *SIAM Journal on Computing*, 39(3):1088–1106, 2009.
32. Dan Garber. Efficient online linear optimization with approximation algorithms. In *Advances in Neural Information Processing Systems*, pages 627–635, 2017.
33. Elad Hazan, Wei Hu, Yuanzhi Li, and Zhiyuan Li. Online improper learning with an approximation oracle. *arXiv preprint arXiv:1804.07837*, 2018.