When Combinatorial Thompson Sampling meets Approximation Regret

Anonymous Author(s)
Affiliation
Address
`email`

Abstract

We study the behavior of the Combinatorial Thompson Sampling policy (CTS) for combinatorial multi-armed bandit problems (CMAB), within an *approximation regret* setting. Although CTS has attracted a lot of interest, it has a drawback that other usual CMAB policies do not have when considering non-exact oracles: for some oracles, CTS has a poor approximation regret (scaling linearly with the time horizon T) [Wang and Chen, 2018]. A study is then necessary to discriminate the oracles on which CTS could learn. This study was started by Kong et al. [2021]: they gave the first approximation regret analysis of CTS for the *greedy oracle*, obtaining an upper bound of order $\mathcal{O}\big(\log(T)/\Delta^2\big)$, where Δ is some minimal reward gap. In this paper, our objective is to push this study further than the simple case of the greedy oracle. We provide the first $\mathcal{O}(\log(T)/\Delta)$ approximation regret upper bound for CTS, obtained under a specific condition on the approximation oracle, allowing a reduction to the exact oracle analysis. We thus term this condition REDUCE2EXACT, and observe that it is satisfied in many *concrete* examples. In particular, it can be extended to the *probabilistically triggered arms* setting, thus capturing even more problems, such as *online influence maximization*.

1 Introduction

Stochastic multi-armed bandits (MAB) Robbins [1952], Berry and Fristedt [1985], Lai and Robbins [1985] are decision-making problems in which an *agent* acts sequentially in an uncertain environment. At each round $t \in \mathbb{N}^*$, the agent must select one arm from a fixed set of n arms, denoted by $[n] \triangleq \{1, \ldots, n\}$, using a *policy*, based on the feedback from the previous rounds. Then it gets as feedback an *outcome* $X_{i,t} \in \mathbb{R}$ — a random variable sampled from $\mathbb{P}_{X_i}$, independently from previous rounds — where i is the selected arm and $\mathbb{P}_{X_i}$ is a probability distribution — unknown to the agent — of mean μ_i^*. The goal for the agent is to maximize the *cumulative reward* over a total of T rounds (T is the *time horizon* and may be unknown). The performance metric of a policy is its *regret* R_T, which is the expectation of the difference over T rounds between the cumulative reward of the policy that always picked the arm with the highest expected reward and the cumulative reward of the learning policy. MAB models the so called dilemma between exploration and exploitation, i.e., whether to continue exploring arms to obtain more information (and thus strengthen the confidence in the estimates of the distributions $\mathbb{P}_{X_i}$), or to use the information gathered by playing the best arm according to the observations so far.

In this paper, we study stochastic combinatorial multi-armed bandit (CMAB), with *semi-bandit feedback*, a.k.a. stochastic semi-bandit (still abbreviated as CMAB here), an extension of MAB where the agent plays an *action* (also called *super-arm*) $A_t \in \mathcal{A} \subset \mathcal{P}([n])$ at each round t, where $\mathcal{A}$ is fixed and called *action space*. The feedback includes the outcomes of all base arms in the played super-arm.[1] The expected reward, given A_t, is assumed to be in the form[2] $r(A_t, \boldsymbol{\mu}^*)$ (traditionally, the reward is linear and equal to $\mathbf{e}_{A_t}^\intercal \boldsymbol{\mu}^*$). In recent years, CMAB has attracted a lot of interest (see e.g. Cesa-Bianchi and Lugosi [2012], Gai et al. [2012], Chen et al. [2013, 2016], Kveton et al. [2015], Wang and Chen [2017], Perrault [2020]), particularly due to its wide applications in network routing, online advertising, recommender system, influence marketing, etc.

Many CMAB policies are based on the *Upper Confidence Bound* (UCB) approach, extending the classical UCB policy [Auer et al., 2002] from MAB to CMAB. This type of approach uses an optimistic estimate $\boldsymbol{\mu}_t$ of $\boldsymbol{\mu}^*$ (i.e., for which the reward function is overestimated), lying in a well-chosen confidence region. Then, the action is chosen by plugging $\boldsymbol{\mu}_t$ inside an *oracle* (typically, $\mathrm{Oracle}(\boldsymbol{\mu}^*)$ is a maximizer of the reward function $A \mapsto r(A, \boldsymbol{\mu}^*)$). An example of such policy is *Combinatorial Upper Confidence Bound* (CUCB) [Chen et al., 2013, Kveton et al., 2015], that uses a Cartesian product of the individual confidence intervals of each arm as a confidence region. For mutually independent arms, Combes et al. [2015] provided the UCB-style policy *Efficient Sampling for Combinatorial Bandit* (ESCB), building a tighter axis-aligned ellipsoidal confidence region around the empirical mean, which helps to better restrict the exploration. Degenne and Perchet [2016] provided a policy called OLS-UCB, leveraging a *sub-Gaussianity* assumption on the arms to generalize the ESCB approach. These policies have been further extended to more general settings afterwards [Perrault et al., 2020b,c, 2019]. Although improving CUCB, all these generalizations are inefficient in terms of computation time.

Another paradigm that has recently gained interest (and which will be our focus in this paper) is to rely on *Thompson Sampling* (TS) instead of UCB. Although introduced much earlier by Thompson [1933], the theoretical analysis of TS for MAB is quite recent: Kaufmann et al. [2012], Agrawal and Goyal [2012] gave a regret bound matching the UCB policy theoretically. Moreover, TS often performs better than UCB in practice, making TS an attractive policy for further investigations. For CMAB, TS extends to *Combinatorial Thompson Sampling* (CTS). In CTS, the unknown mean $\boldsymbol{\mu}^*$ is associated with a belief (a prior distribution, that could be e.g. a product of Beta or Gaussian distributions) updated to a posterior with the Bayes'rule, each time a feedback is received. In order to choose an action at round t, CTS draws a sample $\boldsymbol{\theta}_t$ from the current belief, and plays the action given by $\mathrm{Oracle}(\boldsymbol{\theta}_t)$. CTS is an attractive policy because it has similar advantages to the previously mentioned policies working with ellipsoidal confidence regions while being, like CUCB, computationally efficient. Indeed, recently, for mutually independent arms and sub-Gaussian arms respectively, Wang and Chen [2018], Perrault et al. [2020a] proposed tight analyses of CTS.

Unlike UCB-based policies, the analysis of CTS is valid only when Oracle is exact, i.e., when

$$\mathrm{Oracle}(\boldsymbol{\mu}) \in \operatorname*{arg\,max}_{A \in \mathcal{A}} r(A, \boldsymbol{\mu}).$$

Although this holds true for many combinatorial problem described by the pair $(r, \mathcal{A})$ (we recall that r is usually linear), there exist some problems where the requirement on Oracle has to be relaxed in order to make it tractable. This is usually done considering an α-*approximation* oracle [Chen et al., 2013, 2016, Wen et al., 2016], for $\alpha \in (0, 1)$:

$$r(\mathrm{Oracle}(\boldsymbol{\mu}), \boldsymbol{\mu}) \geq \alpha \max_{A \in \mathcal{A}} r(A, \boldsymbol{\mu}). \tag{1}$$

Under an α-approximation oracle, the benchmark cumulative reward is the α-fraction of the optimal reward, leading to the notion of *approximation regret* [Kakade et al., 2009, Streeter and Golovin, 2009, Chen et al., 2016].

Aware of the limitation of their CTS analysis (that works only with exact oracles), Wang and Chen [2018] also proved, in their Theorem 2, that this limitation is not a technical artifact. More precisely, they provided a specific CMAB instance with an associated approximation oracle, such that CTS on this instance and with this oracle must have a regret scaling linearly in T. Although this negative result is of great interest to the research community, some concerns limit its consideration. Indeed, not only the CMAB instance provided by Wang and Chen [2018] is actually a MAB one (meaning

[1]Note that we will consider the *probabilistically triggered arms* extension in this paper, i.e., where the feedback is on triggered arms. For brevity, we do not present this generalization in the introduction.

[2]Henceforth, we typeset vectors and matrices in bold and indicate components with indices, e.g., $\mathbf{a} = (a_i)_{i \in [n]} \in \mathbb{R}^n$. We also let $\mathbf{e}_i$ be the i^{th} canonical unit vector of $\mathbb{R}^n$, and define the *incidence vector* of any subset $A \subset [n]$ as $\mathbf{e}_A \triangleq \sum_{i \in A} \mathbf{e}_i$. We denote by $\mathbf{a} \odot \mathbf{b} \triangleq (a_i b_i)$ the Hadamard product of two vectors $\mathbf{a}$ and $\mathbf{b}$.

that there is an efficient oracle that simply enumerates the arms), so the use of an approximation regret is not justified, but above all, both the oracle and the instance are uncommon and designed for the proof.

Interested in the question of whether the example provided by Wang and Chen [2018] is pathological or generalizable, Kong et al. [2021] recently initiated a study, which revealed that linear approximation regret for CTS seems to be pathological. More precisely, they obtained a $\mathcal{O}\big(\log(T)/\Delta^2\big)$ bound for the specific case of a *greedy* oracle[3], where Δ is some reward gap. Notably, they obtained this result by bounding the approximation regret by a *greedy regret*, that simply replaces $\alpha \max_{A\in\mathcal{A}} r(A, \boldsymbol{\mu}^*)$ with $r(\mathrm{Oracle}(\boldsymbol{\mu}^*), \boldsymbol{\mu}^*)$, using equation (1). They also gave a tight lower bound on the greedy regret.

In this paper, we want to explore another class of oracles covering more problems in practice. Our goal is to demonstrate that although there are instrumental examples of problems where CTS has a linear regret, the majority of concrete problems do not follow this regime, and are in fact similar to the exact oracle case.

Contributions With a specific condition on Oracle, we describe a general set of CMAB problems where the approximation regret of CTS has a $\mathcal{O}(\log(T)/\Delta)$ bound, improving by a factor $1/\Delta$ over the bound of Kong et al. [2021]. This does not contradict their lower bound, which is focused on the greedy regret. We call this set of CMAB problems REDUCE2EXACT, because, as we will see, a REDUCE2EXACT problem can be approximated using a reduction to exact subproblems. We note that REDUCE2EXACT is compatible with the *probabilistically triggered arms* setting, which allows us to capture even more problems, such as online influence maximization [Wen et al., 2017, Wang and Chen, 2017]. We want to focus on concrete problems, so we provide several real-world examples that belongs to REDUCE2EXACT. In particular, REDUCE2EXACT includes concrete problems with a greedy oracle, which was the focus of Kong et al. [2021].

Further related work We refer the reader to Wang and Chen [2018] for more related work on TS for combinatorial bandits. Briefly, one can mention Gopalan et al. [2014], that gave a frequentist high-probability regret bounds for TS with a general action space and feedback model — Komiyama et al. [2015], that studied TS for the m-sets action space — Wen et al. [2015], that studied TS for contextual CMAB problems, using the Bayesian regret metric (see also Russo and Van Roy [2016]).

Other known limitations of CTS Apart from the limitation related to the approximation regret that interests us in this paper, there are some other existing limitations of CTS highlighted in the literature, which we review here: The CTS policy has an exponential constant term in its regret upper bound [Wang and Chen, 2018, Perrault et al., 2020a], and Wang and Chen [2018] proved in their Theorem 3 that this is unavoidable. A similar behavior have been demonstrated in Zhang and Combes [2021], where it is shown that CTS does not scale polynomially in the ambient dimension n in general. In addition, Zhang and Combes [2021] also proved that CTS is not minimax optimal.

The strengths of CTS Despite the weaknesses mentioned above, CTS remains a widely used policy, mainly because of its empirical performance. Indeed, CTS generally outperforms other policies such as CUCB and ESCB [Wang and Chen, 2018, Perrault et al., 2020a]. Moreover, it is relatively simple to implement, and is computationally efficient (just like CUCB). On the theory side, another advantage is that for an exact oracle, CTS is asymptotically quasi-optimal[4] for many settings where CUCB is not, and where ESCB is computationally inefficient [Perrault et al., 2020a]. It would be desirable that these advantages also apply to the case of approximation regret, thus motivating our investigations.

2 Model and definitions

For more generality, we consider the *probabilistically triggered arms* extension of CMAB [Chen et al., 2016, Wang and Chen, 2017], abbreviated to CMAB-T. In this context, the action $A \in \mathcal{A}$ selected is not necessarily equal to the triggered super-arm S. More precisely, the action space $\mathcal{A}$ is no longer necessarily a subset of $\mathcal{P}([n])$, and can be infinite. At round t, the agent selects $A_t \in \mathcal{A}$,

[3]It is worth mentioning that this oracle is one of the most common, so it is logical to focus on it first.

[4]This means that it has a distribution-dependent regret upper bound whose leading term in T has an optimal rate, up to a poly-logarithmic factor in n.

based on the history of observations $\mathcal{H}_t \triangleq \sigma\big(\mathbf{X}_1 \odot \mathbf{e}_{S_1}, \ldots, \mathbf{X}_{t-1} \odot \mathbf{e}_{S_{t-1}}\big)$ and a possible extra source of randomness (we denote by $\mathcal{F}_t$ the filtration containing $\mathcal{H}_t$ and the extra randomness of round t — in particular, $A_t \in \mathcal{F}_t$). Then, an independent sample $\mathbf{X}_t \sim \mathbb{P}_{\mathbf{X}},\ \mathbf{X}_t \in \mathbb{R}^n$ is drawn and a random subset $S_t \in \mathcal{S} \subset \mathcal{P}([n])$ of arms are triggered ($\mathcal{S}$ is called *super-arm space* or *subset space*). We assume that S_t is drawn independently from a distribution $D_{\text{trig}}(A_t, \mathbf{X}_t)$. For the feedback, the outcomes of each triggered arm is observed, i.e., $\mathbf{e}_{S_t} \odot \mathbf{X}_t$ is observed. The expected reward is of the form $r(A_t, \boldsymbol{\mu}^*)$, where r is a function defined on a domain $\mathcal{A} \times \mathcal{M}$, with $\mathcal{M} \subset \mathbb{R}^n$. Quantities $\mathcal{A}, D_{\text{trig}}, r$ are known to the agent. Under an α-approximation Oracle, we use the approximation regret to evaluate the performance of a policy π, defined as follows.

Definition 1 (Approximation regret)**.** *The T-round α-approximation regret of a learning policy π that selects action $A_t \in \mathcal{A}$ at round t is defined as follows, where the approximation gap is defined as $\Delta_t = \Delta(A_t) \triangleq 0 \vee (\alpha r(A^*, \boldsymbol{\mu}^*) - r(A_t, \boldsymbol{\mu}^*))$, with $A^* \in \arg\max_{A \in \mathcal{A}} r(A, \boldsymbol{\mu}^*)$.*

$$R_{T,\alpha}(\pi) \triangleq \mathbb{E}\left[\sum_{t \in [T]} \Delta_t\right].$$

To approach the problem of minimizing $R_{T,\alpha}$, assumptions are often made [Wang and Chen, 2017]:

Assumption 1 (Approximation oracle)**.** *The agent has access to an* Oracle *such that for any mean vector $\boldsymbol{\mu} \in \mathcal{M}$, $r(\text{Oracle}(\boldsymbol{\mu}), \boldsymbol{\mu}) \geq \alpha r(A^*, \boldsymbol{\mu})$.*

Assumption 2 (1-norm triggering probability modulated bounded smoothness)**.** *There exists $\mathbf{B} \in \mathbb{R}_+^n$ such that for all $A \in \mathcal{A}$, for all $\boldsymbol{\mu}, \boldsymbol{\mu}' \in \mathcal{M}$,*

$$|r(A, \boldsymbol{\mu}) - r(A, \boldsymbol{\mu}')| \leq \sum_{i \in [n]} p_i(A) B_i |\mu_i - \mu_i'|,$$

where the triggering probabilities are defined as $p_i(A) \triangleq \mathbb{P}[i \in S]$, with $S \sim D_{trig}(A, \mathbf{X})$, $\mathbf{X} \sim \mathbb{P}_{\mathbf{X}}$ (we thus have $\sum_{i \in [n]} p_i(A) B_i |\mu_i - \mu_i'| = \mathbb{E}[\|\mathbf{e}_S \odot \mathbf{B} \odot (\boldsymbol{\mu} - \boldsymbol{\mu}')\|_1]$).

Assumption 3 (Sub-Gaussianity of the outcome distribution)**.** *$\mathbb{P}_{\mathbf{X}}$ is such that $\forall \boldsymbol{\lambda} \in \mathbb{R}^n$,*

$$\mathbb{E}\left[e^{\boldsymbol{\lambda}^\mathsf{T}(\mathbf{X} - \boldsymbol{\mu}^*)}\right] \leq e^{\|\boldsymbol{\lambda}\|_2^2/8}.$$

For example, $\mathbb{P}_{\mathbf{X}} = \otimes_{i \in [n]} \mathbb{P}_{X_i}$, and $X_i \overset{a.s.}{\in} [0,1]$ (from Hoeffding's Lemma [Hoeffding, 1963]).

Definition 2 (Other useful definitions)**.** *We also define:* • $B \triangleq \max_{i \in [n]} B_i$, • $\forall i \in [n],\ \Delta_{i,\min} \triangleq \inf_{A \in \mathcal{A}:\ p_i(A) > 0,\ \Delta(A) > 0} \Delta(A)$, • $\Delta_{\min} \triangleq \min_{i \in [n]} \Delta_{i,\min}$, • $\Delta_{\max} \triangleq \max_{A \in \mathcal{A}} \Delta(A)$, • *For* $A \in \mathcal{A}$, $\widetilde{A} \triangleq \{i \in [n] : p_i(A) > 0\}$, • $m \triangleq \max_{A \in \mathcal{A}} \big|\widetilde{A}\big|$, • $m^* \triangleq \sum_{i \in [n]} \mathbb{I}\{p_i(A^*) > 0\} = \big|\widetilde{A^*}\big|$, • $p^* \triangleq \min_{i \in [n]:\ A \in \mathcal{A}\ p_i(A) > 0} p_i(A)$.

3 Combinatorial Thompson Sampling and exact regret analysis

In this section, we present the CTS policy, focusing on two versions, one working with a Beta prior, and the other with a Gaussian prior. Then, we present an associated analysis for the exact regret (i.e., with $\alpha = 1$).

3.1 Algorithms

Based on the above assumptions, we focus on two versions of CTS. The first version is CTS-BETA [Wang and Chen, 2018] (Algorithm 1), working when we assume furthermore that $\mathbb{P}_{\mathbf{X}} = \otimes_{i \in [n]} \mathbb{P}_{X_i}$ and $\mathbf{X} \overset{a.s.}{\in} [0,1]^n$. For each arm $i \in [n]$, CTS-BETA maintains a Beta prior distribution with parameters γ_i and δ_i (initialized to 1). At each round t, for each arm i, the algorithm sample $\theta_{i,t}$ from the corresponding prior, representing the current estimate of μ_i^*. Then the oracle outputs the action A_t to play according to the input vector $\boldsymbol{\theta}_t$. Based on the observation feedback, the algorithm then updates the corresponding Beta distributions. The second version is CTS-GAUSSIAN [Perrault et al., 2020a] (Algorithm 2), that works under the more general Assumption 3. It is essentially the same as

Algorithm 1 CTS-BETA

Initialization: For each arm i, let $\gamma_i = \delta_i = 1$.
For all $t \geq 1$:
  Draw $\boldsymbol{\theta}_t \sim \otimes_{i\in[n]}\text{Beta}(\gamma_i, \delta_i)$.
  Play $A_t = \text{Oracle}(\boldsymbol{\theta}_t)$.
  Get the observation $\mathbf{X}_t \odot \mathbf{e}_{S_t}$, and draw $\mathbf{Y}_t \sim \otimes_{i\in S_t}\text{Bernoulli}(X_{i,t})$.
  For all $i \in S_t$ update $\gamma_i \leftarrow \gamma_i + Y_{i,t}$ and $\delta_i \leftarrow \delta_i + 1 - Y_{i,t}$.

Algorithm 2 CTS-GAUSSIAN

Input: $\beta > 1$.
Initialization: Play each arm once (if the agent knows that $\boldsymbol{\mu}^* \in [a,b]^n$, this might be skipped)
For every subsequent round t:
  Draw $\boldsymbol{\theta}_t \sim \otimes_{i\in[n]}\mathcal{N}\big(\overline{\mu}_{i,t-1}, N_{i,t-1}^{-1}\beta/4\big)$ ($\theta_{i,t} \sim \mathcal{U}[a,b]$ if $N_{i,t-1} = 0$).
  Play $A_t = \text{Oracle}(\boldsymbol{\theta}_t)$.
  Get the observations $\mathbf{X}_t \odot \mathbf{e}_{S_t}$ and let $\mathbf{Y}_t = \mathbf{X}_t$.
  Update $\overline{\boldsymbol{\mu}}_{t-1}$ and counters accordingly.

CTS-BETA, except that the prior distributions are Gaussian. For both algorithms, and an arm $i \in [n]$, we define the number of time i has been triggered at the beginning of round t, called counter of arm i, as $N_{i,t-1} \triangleq \sum_{t'\in[t-1]} \mathbb{I}\{i \in S_{t'}\}$. We also define the *empirical mean* at the beginning of round t as $\overline{\mu}_{i,t-1} \triangleq \sum_{t'\in[t-1]} \frac{\mathbb{I}\{i\in S_{t'}\}Y_{i,t'}}{N_{i,t-1}}$.

3.2 Analysis of CTS: the $\alpha = 1$ case

Although this is close to some known results in the current literature [Huyuk and Tekin, 2019, Perrault et al., 2020a], there is no proof for the classical $\mathcal{O}\Big(\log^2(m)\log(T)\sum_{i\in[n]} B_i^2/\Delta_{i,\min}\Big)$ regret bound under the above assumptions in the CMAB-T setting. We thus provide such a result in Theorem 1 (the proof is postponed to Appendix A). We can notice a notable difference with the work of Huyuk and Tekin [2019] concerning the Assumption 2, where the triggering probabilities do not appear (and are present in their final regret bound). The main difference with Perrault et al. [2020a] is that they do not consider probabilistically triggered arms.

Theorem 1. *For $\alpha = 1$, the policy π described in Algorithm 1 (under Assumptions 1, 2 and $\mathbb{P}_{\mathbf{X}} = \otimes_{i\in[n]}\mathbb{P}_{X_i}$, $\mathbf{X} \overset{a.s.}{\in} [0,1]^n$) or Algorithm 2 (under Assumptions 1, 2 and 3) has regret of order*

$$R_{T,1}(\pi) = \mathcal{O}\left(\sum_{i\in[n]} \frac{B_i^2 \log^2(m)\log(T)}{\Delta_{i,\min}}\right).$$

In addition to being a new result in itself, Theorem 1 will be useful for the $\alpha < 1$ case.

4 The $\alpha < 1$ case for REDUCE2EXACT problems

We will now look at the $\alpha < 1$ case. Our strategy is based on the following observation: many approximation algorithms involve a relaxation, or a reduction to one or more problems that can be solved exactly. Thus, just as the approximation relation is obtained by linking the original problem to exact subproblems, here we want to link the approximation regret, in a similar way, to exact subregrets. We formalize this idea through the following assumption.

Assumption 4 (REDUCE2EXACT)**.** Oracle *is of the form* Oracle $=$ Oracle$_2 \circ$ Oracle$_1$. *The aim of* Oracle$_1$ *is to solve one or more exact subproblems, while the aim of* Oracle$_2$ *is to build an action for the original approximation problem using the intermediate solutions provided by* Oracle$_1$. *Formally:*
• Oracle$_1$: *There are ℓ reward functions $(r_j)_{j\in[\ell]}$. For $\boldsymbol{\mu} \in \mathcal{M}$,* Oracle$_1(\boldsymbol{\mu})$ *is a sequence of exact subactions $(E_j)_{j\in[\ell]}$: for all $j \in [\ell]$, E_j belongs to a subaction space $\mathcal{E}_j$, $\mathcal{E}_j$ can depends on $E_1, \ldots, E_{j-1}$ (the subproblems can be nested) and $E_j \in \arg\max_{E\in\mathcal{E}_j} r_j(E, \boldsymbol{\mu})$.*

• Oracle_2 : *There is some vector* $\mathbf{c} \in \mathbb{R}_+^\ell$ *such that the following is true. We fix the input as* $E_1 \in \mathcal{E}_1, \ldots, E_\ell \in \mathcal{E}_\ell$*. For all* $j \in [\ell]$*, let* $E_j^* \in \arg\max_{E \in \mathcal{E}_j} r_j(E, \boldsymbol{\mu}^*)$*. We have:*

$$\Delta\big(\text{Oracle}_2\big((E_j)_{j\in[\ell]}\big)\big) \leq \sum_{j\in[\ell]} 0 \vee \big(r_j\big(E_j^*, \boldsymbol{\mu}^*\big) - r_j(E_j, \boldsymbol{\mu}^*)\big) \cdot c_j. \tag{2}$$

Each r_j *must satisfy Assumption 2 with some constant* $\mathbf{B}_j \in \mathbb{R}_+^n$ *and with the triggering probabilities* $p_i\big(\text{Oracle}_2\big((E_j)_{j\in[\ell]}\big)\big)$*.*

At first sight, this Assumption 4 seems very specific and rather difficult to fulfill. However, we will see in subsection 4.2 that many problems satisfy it.

4.1 Analysis

In this section, we give in Theorem 2 the main result of this paper. It basically states that under Assumptions 4, the CTS policy have a regret bound comparable to the exact oracle case. The proof is postponed to Appendix B.

Theorem 2. *The policy* π *described in Algorithm 1 (under Assumptions 4 and* $\mathbb{P}_{\mathbf{X}} = \otimes_{i\in[n]}\mathbb{P}_{X_i}$*,* $\mathbf{X} \overset{a.s.}{\in} [0,1]^n$*) or Algorithm 2 (under Assumptions 3 and 4) has regret of order*

$$R_{T,\alpha}(\pi) = \mathcal{O}\left(\sum_{i\in[n]}\sum_{j\in[\ell]} \frac{c_j^2 B_{ij}^2 \log^2(m)\log(T)}{\Delta_{i,\min}}\right).$$

The idea of the proof is quite simple once Assumption 4 has been made. We can see that the approximation regret can be decomposed into ℓ exact sub-regrets, according to equation (2). The point to note with these sub-regrets is that they can be nested and depend on each other, so we must be careful handling them.

4.2 Examples of REDUCE2EXACT problems

Here, we present several problems belonging to REDUCE2EXACT. Each time, after a quick introduction of the problem, we translate it into our CMAB-T context, and finally show how it satisfies the REDUCE2EXACT criteria.

Probabilistic maximum coverage (PMC) This is one of the main examples proposed by Kong et al. [2021]. Given a weighted bipartite graph $G = (L, R, E)$, with weights $\boldsymbol{\mu}^* \triangleq (\mu^*_{(u,v)})_{(u,v)\in E}$, the goal is to find an action $A \in \mathcal{A} \triangleq \{A \subset L : |A| = k\}$ maximizing the expected number of influenced nodes in R, where each node $v \in R$ can be independently influenced by $u \in A$ with probability $\mu^*_{(u,v)}$, i.e., maximizing $f(A, \boldsymbol{\mu}^*) \triangleq \sum_{v\in R}\Big(1 - \prod_{u\in A:\,(u,v)\in E}(1 - \mu^*_{(u,v)})\Big)$. This problem can be applied to the semi-bandit framework called *the ad placement problem*, where L are the web pages, R are the users and $\mu^*_{(u,v)}$ is the probability that user v clicks on the ad on web page u. In this application, the user's click probabilities are unknown and must be learned as the rounds progress. The Greedy oracle can provide an approximate solution with approximation ratio $\alpha = 1 - 1/e$ [Nemhauser et al., 1978]. It fits REDUCE2EXACT as follows:

• $\text{Oracle}_1(\boldsymbol{\mu})$: For $i \in [k]$, we define $\mathcal{E}_i \triangleq \{(a_1, \ldots, a_i) : (a_1, \ldots, a_{i-1}) = E_{i-1}, a_i \in L \backslash E_{i-1}\}$ and $r_i((a_1, \ldots, a_i), \boldsymbol{\mu}) \triangleq f(\{a_1, \ldots, a_i\}, \boldsymbol{\mu})$.

• $\text{Oracle}_2\big((E_j)_{j\in[k]}\big)$: Let $(a_1, \ldots, a_k) \triangleq E_k$. Oracle_2 returns $A = \{a_1, \ldots, a_k\}$. Let $A^i = \{a_1, \ldots, a_i\}$ for some $i \in [k]$ and by abuse of notation, let $f = f(\ \cdot\ , \boldsymbol{\mu}^*)$. The following is true using that f is monotone submodular[5] (this is actually the way Nemhauser et al. [1978] proved the approximation guarantee, and is true for any monotone submodular function):

$$f(A^*) - f(A^i) \leq \sum_{a\in A^*\backslash A^i} \big(f(\{a\}\cup A^i) - f(A^{i+1})\big) + k\big(f(A^{i+1}) - f(A^i)\big).$$

[5] f is monotone if for every $A \subset B$, we have $f(A) \leq f(B)$. It is submodular if for every A, B we have that $f(A\cup B) + f(A\cap B) \leq f(A) + f(B)$.

Once the above relation is obtained, we can actually continue the original proof from Nemhauser et al. [1978], skipping each step where we would need to use property of $\mathrm{Oracle}_1(\boldsymbol{\mu}^*)$, thus leaving a term in the right-hand side for each time we skipped.

$$\begin{aligned}
&f(A^*) - f(A^k) \\
&= f(A^*) - f(A^{k-1}) - \big(f(A^k) - f(A^{k-1})\big) \\
&\leq \big(f(A^*) - f(A^{k-1})\big)\left(1 - \frac{1}{k}\right) + \frac{1}{k}\sum_{a \in A^* \setminus A^{k-1}} \big(f(\{a\} \cup A^{k-1}) - f(A^k)\big) \\
\ldots &\leq f(A^*)\left(1 - \frac{1}{k}\right)^k + \sum_{i=1}^{k} \frac{\left(1 - \frac{1}{k}\right)^{i-1}}{k} \sum_{a \in A^* \setminus A^{k-i}} \big(f(\{a\} \cup A^{k-i}) - f(A^{k-i+1})\big).
\end{aligned}$$

Finally, since $\left(1 - \frac{1}{k}\right)^k \leq e^{-1}$, we get that $\Delta\big((E_j)_{j \in [k]}\big) = (1 - e^{-1})f(A^*) - f(A)$ is bounded by

$$\sum_{i=1}^{k} \frac{\left(1 - \frac{1}{k}\right)^{i-1}}{k} \sum_{a \in A^* \setminus A^{k-i}} \big(r_{k-i+1}(E^*_{k-i+1}, \boldsymbol{\mu}^*) - r_{k-i+1}(E_{k-i+1}, \boldsymbol{\mu}^*)\big).$$

Note in passing that we recover the classical approximation if the right-hand-side was bounded by 0. It is proved that the reward function satisfies Assumption 2 [Chen et al., 2016], which directly implies that each r_j satisfies Assumption 2.

Online influence maximization (OIM) As the analysis mentioned above only uses submodularity, it can be extended to the problem of online influence maximization in a social network. A social network is modeled as a directed graph $G = (V, E)$, with nodes V representing users and edges E representing connections. For a node $i \in V$, a subset $A \subset V$, and a vector $\mathbf{x} \in \{0,1\}^E$, the predicate $A \overset{\mathbf{x}}{\rightsquigarrow} i$ holds if, in the graph defined by $G_{\mathbf{x}} \triangleq (V, \{ij \in E, x_{ij} = 1\})$, there is a forward path from a node in A to the node i. If it holds, we say that i is influenced by A under $\mathbf{x}$. The goal is to find an action $A \in \mathcal{A} \triangleq \{A \subset V : \ |A| = k\}$ maximizing $\sigma(A, \boldsymbol{\mu}^*) \triangleq \mathbb{E}\Big[\Big|\Big\{i \in V,\ A \overset{\mathbf{X}}{\rightsquigarrow} i\Big\}\Big|\Big]$, where $\mathbf{X} \sim \otimes_{(u,v)\in E} \mathrm{Bernoulli}(\mu^*_{(u,v)})$. This model is called the independent cascade model. A notable property to use the greedy oracle is that σ is monotone submodular. As the exact calculation of σ is prohibitive, it is estimated by a Monte Carlo simulation, resulting in an approximation factor $\alpha = 1 - e^{-1} - \varepsilon$ [Feige, 1998, Chen et al., 2010] in the above greedy oracle analysis.

Metric k-center Given a set of cities, one wants to build k warehouses in different cities and minimize the maximum distance of a city to a warehouse. Formally, given a complete undirected weighted graph $G = (V, E)$ whose distances $d(v_i, v_j)$ satisfy the triangle inequality, the goal is to find an action $A \in \mathcal{A} \triangleq \{A \subset V : \ |A| = k\}$ that minimizes $\max_{v \in V} d(v, A)$. We can consider the semi-bandit setting where the set of base arms is E, $\boldsymbol{\mu}^* \triangleq (d(v_i, v_j))_{(v_i, v_j) \in E}$ and the feedback set S includes the edges of the graph induced by the chosen action. We can target an approximation regret with $\alpha = 1/2$ using the following oracle (which is simply the standard greedy algorithm).

• $\mathrm{Oracle}_1(\boldsymbol{\mu})$: For $i \in [k]$, we define $\mathcal{E}_i \triangleq \{(a_1, \ldots, a_i) : (a_1, \ldots, a_{i-1}) = E_{i-1}, a_i \in V \backslash E_{i-1}\}$ and $r_i((a_1, \ldots, a_i), \boldsymbol{\mu}) \triangleq \min_{j \in [i-1]} \mu_{a_i, a_j}$ (with $r_1 = 0$). One can notice we thus have $E_i^* \in \arg\max_{(a_1, \ldots, a_i) \in \mathcal{E}_i} d(a_i, \{a_1, \ldots, a_{i-1}\}) = \arg\max_{(a_1, \ldots, a_i) \in \mathcal{E}_i} d(a_i, E_{i-1})$.

• $\mathrm{Oracle}_2\big((E_j)_{j \in [k]}\big)$: Let $(a_1, \ldots, a_k) \triangleq E_k$. Oracle_2 returns $A = \{a_1, \ldots, a_k\}$. Let $w \in \arg\max_{v \in V} d(v, A)$. We thus have:

$$\begin{aligned}
\frac{1}{2}\max_{v \in V} d(v, A) - \max_{v \in V} d(v, A^*) &\leq \frac{1}{2}\left(\max_{v \in V} d(v, A) - \min_{a \in A \cup \{w\}} \min_{a' \in A \backslash \{a\}} d(a, a')\right) \\
&= \frac{1}{2}\max_{j \in [k-1]}\left(\max_{v \in V} d(v, A) - d(a_{j+1}, E_j)\right) \\
&\leq \frac{1}{2}\max_{j \in [k-1]}\left(\max_{v \in V} d(v, E_j) - d(a_{j+1}, E_j)\right) \\
&= \frac{1}{2}\max_{j \in [k-1]}\big(r_{j+1}(E^*_{j+1}, \boldsymbol{\mu}^*) - r_{j+1}(E_{j+1}, \boldsymbol{\mu}^*)\big).
\end{aligned}$$

Where the first inequality is deduced as follows: the map $f: v \mapsto \arg\min_{a^* \in A^*} d(a^*, v)$ defines a partition of V into $k = |A^*|$ clusters. By the the pigeonhole principle, one cluster contains 2 different points $a, a' \in A \cup \{w\}$ (simply because its size is $k+1$). We can assume $a' \in A$ without loss of generality. Thus, since $f(a) = f(a')$, we get $d(a, a') \leq d(a, f(a)) + d(a', f(a')) = d(a, A^*) + d(a', A^*) \leq 2 \max_{v \in V} d(v, A^*)$.

Vertex cover The problem consists, given an undirected graph $G = (V, E)$, in finding a set of vertices with minimal cost to cover all the edges of E. Formally, with $\boldsymbol{\mu}^* \in \mathbb{R}_+^V$, the goal is to find an action $A \in \mathcal{A} \triangleq \{A \subset V : \forall (u, v) \in E, u \in A \text{ or } v \in A\}$ that minimizes $\mathbf{e}_A^\intercal \boldsymbol{\mu}^*$. The semi-bandit feedback is defined directly as $S = A$. We can target an approximation regret with $\alpha = 1/2$ using the following linear programming (LP) relaxation oracle.

• $\text{Oracle}_1(\boldsymbol{\mu})$: Let[6] $\mathcal{E}_1 \triangleq \left\{\mathbf{x} \in \{0, 1/2, 1\}^V : \forall (u, v) \in E,\ x_u + x_v \geq 1\right\}$ and $r_1(\mathbf{x}, \boldsymbol{\mu}) \triangleq -\mathbf{x}^\intercal \boldsymbol{\mu}$.

• $\text{Oracle}_2(E_1)$: Let $\mathbf{x} \triangleq E_1$. Oracle_2 returns $A = \{v \in V : x_v \geq 1/2\}$, that belongs to $\mathcal{A}$ since $\forall\ (u, v) \in E, x_u + x_v \geq 1$, so $x_u \geq 1/2$ or $x_v \geq 1/2$, so $u \in A$ or $v \in A$. Let $\mathbf{x}^* \triangleq E_1^*$. We have:

$$
\begin{aligned}
\frac{1}{2}\mathbf{e}_A^\intercal \boldsymbol{\mu}^* - \mathbf{e}_{A^*}^\intercal \boldsymbol{\mu}^* &\leq \mathbf{x}^\intercal \boldsymbol{\mu}^* - \mathbf{e}_{A^*}^\intercal \boldsymbol{\mu}^* \\
&\leq \mathbf{x}^\intercal \boldsymbol{\mu}^* - \mathbf{x}^{*\intercal} \boldsymbol{\mu}^* \\
&= r_1(E_1^*, \boldsymbol{\mu}^*) - r_1(E_1, \boldsymbol{\mu}^*).
\end{aligned}
$$

Notice, r_1 satisfies Assumption 2 with the constants being 1, using $\mathbf{x} \in \{0, 1/2, 1\}^V$, so that $\mathbf{x} \leq \mathbf{e}_A$.

Max-Cut Given an undirected weighted graph $G = (V, E)$, with weights $\boldsymbol{\mu}^* \triangleq (\mu^*_{(u,v)})_{(u,v) \in E}$, the goal is to find an action $A \in \mathcal{A} \triangleq \{A \subset V\}$ maximizing the total weight of the edges between A and its complement, i.e., $\frac{1}{2}\sum_{(u,v) \in E} \mu^*_{(u,v)}(1 - y_u y_v)$, where $\mathbf{y} \triangleq \mathbf{e}_A - \mathbf{e}_{V \setminus A}$. We consider the semi-bandit context where E is the set of arms, and the feedback set includes the edges between A and its complement, i.e., $S = \{(u, v) \in E,\ y_u y_v = -1\}$. To include randomization within the oracle, we can extend the action space $\mathcal{A}$ to the set of probability measures on $\{A \subset V\}$, replacing $\Delta(A)$ by its expectation on A, as a function of the distribution of A. The polynomial-time approximation algorithm for Max-Cut with the best known approximation ratio [Goemans and Williamson, 1995] uses semidefinite programming and randomized rounding, and achieves an approximation ratio of $\alpha = \frac{2}{\pi} \min_{0 \leq \theta \leq \pi} \frac{\theta}{1 - \cos\theta} \approx 0.878$. In our context, it can be defined as follows.

• $\text{Oracle}_1(\boldsymbol{\mu})$: We define $\mathcal{E}_1 \triangleq \left\{(\mathbf{v}_u)_{u \in V} \in \left\{\mathbf{v} \in \mathbb{R}^V : \|\mathbf{v}\|_2 = 1\right\}^V\right\}$ and $r_1((\mathbf{v}_u)_{u \in V}, \boldsymbol{\mu}) \triangleq \frac{1}{2}\sum_{(u,v) \in E} \mu_{(u,v)}(1 - \mathbf{v}_u^\intercal \mathbf{v}_v)$.

• $\text{Oracle}_2(E_1)$: Let $(\mathbf{v}_u)_{u \in V} \triangleq E_1$. Oracle_2 returns the distribution of $A = \{u \in V,\ \mathbf{v}_u^\intercal \mathbf{Z} \geq 0\}$, where $\mathbf{Z} \sim \mathcal{U}\left(\left\{\mathbf{v} \in \mathbb{R}^V : \|\mathbf{v}\|_2 = 1\right\}\right)$. The following is proved by Goemans and Williamson [1995]:

$$
\begin{aligned}
\sum_{(u,v) \in E} \mu_{(u,v)} \mathbb{P}[(u, v) \in S] &= \mathbb{E}\left[\sum_{(u,v) \in E} \mu_{(u,v)} \frac{1 - \text{sign}(\mathbf{v}_u^\intercal \mathbf{Z})\text{sign}(\mathbf{v}_v^\intercal \mathbf{Z})}{2}\right] \\
&= \sum_{(u,v) \in E} \mu_{(u,v)} \frac{\arccos(\mathbf{v}_u^\intercal \mathbf{v}_v)}{\pi} \\
&\geq \alpha \sum_{(u,v) \in E} \mu_{(u,v)} \frac{1 - \mathbf{v}_u^\intercal \mathbf{v}_v}{2} = \alpha r_1((\mathbf{v}_u)_{u \in V}, \boldsymbol{\mu}).
\end{aligned}
$$

Thus, r_1 satisfies Assumption 2 with the constants being $1/\alpha$. We also get, with $\mathbf{y}^* \triangleq \mathbf{e}_{A^*} - \mathbf{e}_{V \setminus A^*}$:

$$
\begin{aligned}
\Delta(\text{Oracle}_2(E_1)) &= \alpha \sum_{(u,v) \in E} \mu^*_{(u,v)} \frac{1 - y_u^* y_v^*}{2} - \mathbb{E}\left[\sum_{(u,v) \in E} \mu^*_{(u,v)} \frac{1 - \text{sign}(\mathbf{v}_u^\intercal \mathbf{Z})\text{sign}(\mathbf{v}_v^\intercal \mathbf{Z})}{2}\right] \\
&\leq \alpha(r_1(E_1^*, \boldsymbol{\mu}^*) - r_1(E_1, \boldsymbol{\mu}^*)).
\end{aligned}
$$

[6]The LP relaxation of vertex cover is half-integral, so that we can allow each variable to be in $\{0, 1/2, 1\}$ rather than the interval from 0 to 1.

4.3 A counter-example

We give here a case that does not belong to REDUCE2EXACT, while being concrete (compared to the example provided by Wang and Chen [2018]). The goal is to show that some concrete examples do not belong to our oracle class.

Travelling salesman problem (TSP) Given a complete undirected weighted graph $G = (V, E)$ whose distances $\boldsymbol{\mu}^* \triangleq (d(u,v))_{(u,v)\in E}$ satisfy the triangle inequality, the goal is to find an Hamiltonian cycle $A \in \mathcal{A} \triangleq \left\{(v_1, \dots, v_{|V|}) : \left\{v_1, \dots, v_{|V|}\right\} = V\right\}$ of minimum cost $\sum_{i\in[|V|]} d(v_{i-1}, v_i)$, with $v_0 = v_{|V|}$. We consider the following oracle from the Christofides [1976] algorithm ($\alpha = 2/3$).
• $\mathrm{Oracle}_1(\boldsymbol{\mu})$: The algorithm of Christofides and Serdyukov combines $\mathcal{E}_1 \triangleq \{\text{spanning trees of } G\}$ and $\mathcal{E}_2 \triangleq \{\text{perfect matchings of the subgraph } G' \text{ of } G \text{ induced by the vertices of odd order in } E_1\}$, with r_1 being the weight of the spanning tree and r_2 the weight of the perfect matching.
• $\mathrm{Oracle}_2(E_1, E_2)$: Oracle_2 combines the edges of E_1 and E_2 to form a connected multigraph in which all vertices have even degree (so it is Eulerian), forms an Eulerian circuit in this multigraph, and finally, outputs the Hamiltonian cycle obtained by skipping repeated vertices (shortcutting). Thanks to the triangle inequality, shortcutting does not increase the weight, so we have

$$\mathbf{e}_A^\intercal \boldsymbol{\mu}^* \leq r_1(E_1, \boldsymbol{\mu}^*) + r_2(E_2, \boldsymbol{\mu}^*).$$

Let's now deal with A^*. Removing an edge from A^* produces a spanning tree, so $\mathbf{e}_{A^*}^\intercal \boldsymbol{\mu}^* \geq r_1(E_1^*, \boldsymbol{\mu}^*)$. On the other hand, by the triangle inequality, the weight of the optimal TSP solution for G' is lower than $\mathbf{e}_{A^*}^\intercal \boldsymbol{\mu}^*$ (visiting more nodes does not, in any case, reduce the total cost). Taking every second edge of this cycle (which is of even length since it has an even number of vertices) we obtain a matching that has a weight less than half the weight of the cycle (if this is not the case we can take the complementary), so $\mathbf{e}_{A^*}^\intercal \boldsymbol{\mu}^*/2 \geq r_2(E_2^*, \boldsymbol{\mu}^*)$. To summarize, we have

$$\frac{2}{3}\mathbf{e}_A^\intercal \boldsymbol{\mu}^* - \mathbf{e}_{A^*}^\intercal \boldsymbol{\mu}^* \leq \frac{2}{3}(r_1(E_1, \boldsymbol{\mu}^*) - r_1(E_1^*, \boldsymbol{\mu}^*) + r_2(E_2, \boldsymbol{\mu}^*) - r_2(E_2^*, \boldsymbol{\mu}^*)).$$

From the above, we can see that all the criteria of REDUCE2EXACT are satisfied, except Assumption 2. Indeed, we assume that we receive feedback from A, while we would need feedback from the larger set $E_1 \cup E_2$. We could probably have foreseen that the TSP would pose a difficulty in our assumption: indeed, among the many operations performed by the oracle to build the final solution, shortcutting is the one that does not imply an optimization, but rather makes the solution feasible. In other words, if we allowed the tour to pass over the same vertex several times, then the TSP would belong to REDUCE2EXACT by skipping the shortcutting step. Although this gives more information, the fact that it is better to go through the same vertex multiple times is rather counterintuitive, suggesting that perhaps another approach can overcome this issue.

5 Conclusion

In this article, our main objective is to further expand the scope of the CTS policy. We not only expand CTS to probabilistically triggered arms (which was one of the open questions in the Kong et al. [2021]), but we also consider a broader class of oracle compatible with CTS. More precisely, we propose a condition, REDUCE2EXACT, which may seem unnatural at first, but which in fact simply expresses that exact sub-problems must be hidden in the original approximation problem, and that the approximation oracle exploit them to output the final solution. Knowing that the majority of approximation algorithms use one or more relaxations to an exact problem (e.g., solving a convex programming relaxation to obtain a fractional solution and then rounding this fractional solution to get a feasible solution), our assumption falls within the range of many CMAB-T settings. From this reduction to exact problems, we naturally obtain the standard tight regret bound $\mathcal{O}(\log(T)/\Delta)$. This is the first tight bound for the approximation regret on non-exact oracles.

As future work, it may be interesting to explore other types of assumption leading to a similar bound. Through the exploration of new assumptions, we can hope to see some counterexamples disappear (like the TSP for our case). Another point deserving further work would be to better understand/reduce the ℓ dependency in the approximation regret bound given in Theorem 2. Finally, we have that the approximation regret is in some way concervative compared to the greedy regret. For a given oracle, we can easily consider the equivalent of the greedy regret for that oracle. An interesting question would then be to extend the work of Kong et al. [2021] to other types of oracle.

References

S. Agrawal and N. Goyal. Thompson Sampling for Contextual Bandits with Linear Payoffs. *CoRR, abs/1209.3352, http://arxiv.org/abs/1209.3352*, sep 2012. URL http://arxiv.org/abs/1209.3352.

P. Auer, N. Cesa-Bianchi, and P. Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2-3):235–256, 2002.

D. A. Berry and B. Fristedt. *Bandit Problems: Sequential Allocation of Experiments*, volume 38 of *Monographs on statistics and applied probability*. Chapman and Hall, 1985.

N. Cesa-Bianchi and G. Lugosi. Combinatorial bandits. In *Journal of Computer and System Sciences*, volume 78, pages 1404–1422, 2012.

W. Chen, C. Wang, and Y. Wang. Scalable influence maximization for prevalent viral marketing in large-scale social networks. In *Knowledge Discovery and Data Mining*, 2010.

W. Chen, Y. Wang, and Y. Yuan. Combinatorial multi-armed bandit: General framework and applications. In S. Dasgupta and D. McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 151–159, Atlanta, Georgia, USA, 2013. PMLR. URL http://proceedings.mlr.press/v28/chen13a.html.

W. Chen, Y. Wang, and Y. Yuan. Combinatorial multi-armed bandit and its extension to probabilistically triggered arms. *Journal of Machine Learning Research*, 17, 2016.

N. Christofides. Worst-case analysis of a new heuristic for the travelling salesman problem. Technical report, Carnegie-Mellon Univ Pittsburgh Pa Management Sciences Research Group, 1976.

R. Combes, M. S. Talebi Mazraeh Shahi, A. Proutiere, and m. lelarge. Combinatorial bandits revisited. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 2116–2124. Curran Associates, Inc., 2015. URL http://papers.nips.cc/paper/5831-combinatorial-bandits-revisited.pdf.

R. Degenne and V. Perchet. Combinatorial semi-bandit with known covariance. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2972–2980. Curran Associates, Inc., 2016. URL http://papers.nips.cc/paper/6137-combinatorial-semi-bandit-with-known-covariance.pdf.

U. Feige. A threshold of ln n for approximating set cover. *Journal of the ACM (JACM)*, 45(4): 634–652, 1998.

Y. Gai, B. Krishnamachari, and R. Jain. Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations. *Transactions on Networking*, 20(5):1466–1478, 2012.

M. X. Goemans and D. P. Williamson. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM (JACM)*, 42(6): 1115–1145, 1995.

A. Gopalan, S. Mannor, and Y. Mansour. Thompson sampling for complex bandit problems. In *International Conference on Machine Learning*, 2014.

W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30, 1963.

A. Huyuk and C. Tekin. Analysis of thompson sampling for combinatorial multi-armed bandit with probabilistically triggered arms. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pages 1322–1330. PMLR, 2019.

S. M. Kakade, A. T. Kalai, and K. Ligett. Playing games with approximation algorithms. *SIAM Journal on Computing*, 39(3):1088–1106, 2009.

E. Kaufmann, N. Korda, and R. Munos. Thompson Sampling: An Asymptotically Optimal Finite Time Analysis. *Algorithmic Learning Theory*, 2012.

J. Komiyama, J. Honda, and H. Nakagawa. Optimal Regret Analysis of Thompson Sampling in Stochastic Multi-armed Bandit Problem with Multiple Plays. jun 2015. URL http://arxiv.

org/abs/1506.00779.

F. Kong, Y. Yang, W. Chen, and S. Li. The hardness analysis of thompson sampling for combinatorial semi-bandits with greedy oracle, 2021.

B. Kveton, Z. Wen, A. Ashkan, and C. Szepesvari. Tight regret bounds for stochastic combinatorial semi-bandits. In *International Conference on Artificial Intelligence and Statistics*, 2015.

T. L. Lai and H. Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–22, 1985.

G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations for maximizing submodular set functions–I. *Mathematical Programming*, 14(1):265–294, 1978.

P. Perrault. *Efficient Learning in Stochastic Combinatorial Semi-Bandits*. Theses, Univeristé Paris-Saclay, Nov. 2020. URL https://tel.archives-ouvertes.fr/tel-03093268.

P. Perrault, V. Perchet, and M. Valko. Exploiting structure of uncertainty for efficient matroid semi-bandits. In K. Chaudhuri and R. Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5123–5132, Long Beach, California, USA, 2019. PMLR. URL http://proceedings.mlr.press/v97/perrault19a.html.

P. Perrault, E. Boursier, V. Perchet, and M. Valko. Statistical efficiency of thompson sampling for combinatorial semi-bandits. *arXiv preprint arXiv:2006.06613*, 2020a.

P. Perrault, J. Healey, Z. Wen, and M. Valko. Budgeted online influence maximization. In *Proceedings of the 37th International Conference on Machine Learning*, pages 6588–6599. 2020b.

P. Perrault, V. Perchet, and M. Valko. Covariance-adapting algorithm for semi-bandits with application to sparse rewards. In *Conference on Learning Theory*, 2020c.

H. Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematics Society*, 58:527–535, 1952.

D. Russo and B. Van Roy. An information-theoretic analysis of thompson sampling. *The Journal of Machine Learning Research*, 17(1):2442–2471, 2016.

M. Streeter and D. Golovin. An online algorithm for maximizing submodular functions. In *Advances in Neural Information Processing Systems*, pages 1577–1584, 2009.

W. R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25:285–294, 1933.

Q. Wang and W. Chen. Improving regret bounds for combinatorial semi-bandits with probabilistically triggered arms and its applications. In *Neural Information Processing Systems*, 2017. URL http://arxiv.org/abs/1703.01610.

S. Wang and W. Chen. Thompson Sampling for Combinatorial Semi-Bandits. mar 2018. URL http://arxiv.org/abs/1803.04623.

Z. Wen, B. Kveton, and A. Ashkan. Efficient learning in large-scale combinatorial semi-bandits. In *International Conference on Machine Learning*, pages 1113–1122, 2015.

Z. Wen, B. Kveton, and M. Valko. Influence maximization with semi-bandit feedback. Technical report, 2016.

Z. Wen, B. Kveton, M. Valko, and S. Vaswani. Online influence maximization under independent cascade model with semi-bandit feedback. In *Neural Information Processing Systems*, 2017.

R. Zhang and R. Combes. On the suboptimality of thompson sampling in high dimensions. *arXiv preprint arXiv:2102.05502*, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See Abstract and section 1.
   (b) Did you describe the limitations of your work? [Yes] See the second paragraph in section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] This work does not have any potential negative societal impacts.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See section 2.
   (b) Did you include complete proofs of all theoretical results? [Yes] See all sections in Appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [N/A]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [N/A]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [N/A]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]