



# Statistically Efficient, Polynomial Time Algorithms for Combinatorial Semi Bandits

Thibaut Cuvelier, Richard Combes, Eric Gourdin

## ► To cite this version:

Thibaut Cuvelier, Richard Combes, Eric Gourdin. Statistically Efficient, Polynomial Time Algorithms for Combinatorial Semi Bandits. Proceedings of the ACM on Measurement and Analysis of Computing Systems , 2021, 5 (9), pp.1-31. 10.1145/3447387 . hal-03162127

**HAL Id: hal-03162127**

**<https://hal.science/hal-03162127>**

Submitted on 8 Mar 2021

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Statistically Efficient, Polynomial Time Algorithms for Combinatorial Semi Bandits

Thibaut Cuvelier, Richard Combes, Eric Gourdin

January 14, 2021

## Abstract

We consider combinatorial semi-bandits over a set of arms  $\mathcal{X} \subset \{0, 1\}^d$  where rewards are uncorrelated across items. For this problem, the algorithm ESCB yields the smallest known regret bound  $R(T) = \mathcal{O}\left(\frac{d(\ln m)^2(\ln T)}{\Delta_{\min}}\right)$ , but it has computational complexity  $\mathcal{O}(|\mathcal{X}|)$  which is typically exponential in  $d$ , and cannot be used in large dimensions. We propose the first algorithm which is both computationally and statistically efficient for this problem with regret  $R(T) = \mathcal{O}\left(\frac{d(\ln m)^2(\ln T)}{\Delta_{\min}}\right)$  and computational complexity  $\mathcal{O}(T\text{poly}(d))$ . Our approach involves carefully designing an approximate version of ESCB with the same regret guarantees, showing that this approximate algorithm can be implemented in time  $\mathcal{O}(T\text{poly}(d))$  by repeatedly maximizing a linear function over  $\mathcal{X}$  subject to a linear budget constraint, and showing how to solve this maximization problems efficiently.

## 1 Introduction

We consider the combinatorial bandit problem with semi-bandit feedback and independent rewards across items. Time is discrete, and at times  $t = 1, \dots, T$  a learner chooses a decision  $x(t) \in \mathcal{X}$ , where  $\mathcal{X} \subset \{0, 1\}^d$  is a combinatorial set that is known to the learner. The learner then receives a reward  $Z^\top(t)x(t)$  and observes a feedback vector  $Y(t) = (x_1(t)Z_1(t), \dots, x_d(t)Z_d(t))$ , where  $Z(t) \in [0, 1]^d$  is a random vector with mean  $\theta \in \mathbb{R}^d$  and whose entries are independent.

The expected reward from decision  $x \in \mathcal{X}$  is  $\theta^\top x$ , and the goal is to maximize the sum of expected rewards, or equivalently to minimize the regret:

$$R(T) = \sum_{t=1}^T \max_{x \in \mathcal{X}} \{\theta^\top x\} - \mathbb{E}(\theta^\top x(t)).$$

The vector  $\theta$  is unknown to the learner; in order to minimize the regret, one must discover the decision  $x \in \mathcal{X}$  maximizing  $\theta^\top x$ , and in turn one must explore enough decisions to obtain sufficient information about  $\theta$ . This problem models a large amount of practically relevant online decision problems such as online shortest path routing, ad-display optimization, and resource allocation.

**Contribution.** In this paper, we propose the first (to the best of our knowledge) algorithm with regret  $R(T) = \mathcal{O}\left(\frac{d(\ln m)^2(\ln T)}{\Delta_{\min}}\right)$  and polynomial computational complexity in the problem dimension  $d$  for a large family of combinatorial sets  $\mathcal{X}$ .

The rest of this paper is organized as follows. In Section 2, we further highlight the model and describe combinatorial sets of interest. In Section 3, we outline the related work on this problem, including state-of-the-art regret bounds and algorithms, and highlight our contribution. In Section 4, we describe the proposed algorithm, and provide regret bounds. In Section 5, we show that our algorithm may be implemented in polynomial time for a large class of combinatorial sets, and analyze its computational complexity in details. In Section 6, we perform numerical experiments to complement. Section 7 concludes the paper.

## 2 Combinatorial Semi-Bandits

### 2.1 Model

As said previously, we consider the following problem. Time is discrete, and at times  $t = 1, \dots, T$  a learner chooses a decision  $x(t) \in \mathcal{X}$ , where  $\mathcal{X} \subset \{0, 1\}^d$  is a combinatorial set which is known to the learner. Set  $\mathcal{X}$  may be any combinatorial set, including the bases of a matroid, the set of paths in some graph, the set of matchings in a bipartite graph, etc. The problem dimension is  $d$ , and we define  $m = \max_{x \in \mathcal{X}} \mathbf{1}^\top x$  the size of the largest decision.

After selecting decision  $x(t)$ , the learner then receives a reward  $Z^\top(t)x(t)$  and observes a feedback vector  $Y(t) = (x_1(t)Z_1(t), \dots, x_d(t)Z_d(t))$ , where  $Z(t) \in [0, 1]^d$  is a random vector.

We assume that  $(Z(t))_t$  are i.i.d. with mean  $\theta \in [0, 1]^d$  and that the entries of  $Z_i(t)$  are independent as well. Vector  $\theta$  is initially unknown to the learner, and must be learnt by repetitively selecting decisions and observing subsequent feedback. For  $i \in \{1, \dots, d\}$ , if  $x_i(t) = 1$ , then the learner obtains a noisy realization of  $\theta_i$  and nothing otherwise, so that decisions must be carefully selected to obtain a good estimate of  $\theta$ . This is the "semi-bandit feedback" model.

Since  $\theta$  is unknown to the learner, decision  $x(t)$  must be selected solely as a function of the feedback information available at time  $t$ , i.e.  $Y(t-1), \dots, Y(1)$ .

The expected reward received by selecting decision  $x \in \mathcal{X}$  is  $\theta^\top x$  (i.e. rewards are linear in the decision), so that  $\theta_i$  represents the amount of reward received by selecting  $x_i = 1$ . The optimal decision is  $x^* \in \arg \max_{x \in \mathcal{X}} \{\theta^\top x\}$  (there may be several optimal decisions). We define the reward gap  $\Delta_x = \theta^\top (x^* - x)$ , i.e. the amount of regret incurred to the learner by selecting decision  $x$  instead of  $x^*$ . We denote by  $\Delta_{\min} = \min_{x: \Delta_x > 0} \Delta_x$  the smallest non-null gap.

The goal of the learner is to minimize the regret, which is simply the difference in terms of expected cumulative rewards between the learner and an oracle who knows the latent vector  $\theta$  in advance and who always selects the optimal decision  $x^*$ , that is:

$$R(T) = \sum_{t=1}^T \mathbb{E}(\Delta_{x(t)}).$$

### 2.2 Combinatorial Sets of Interest

Of course, in order to devise algorithms with low regret and computational complexity, one must take into account the structure of the combinatorial set  $\mathcal{X}$ . We will consider several family of sets, which we present here. We denote by  $e_i$  for  $i = 1, \dots, d$  the  $i$ -th canonical basis vector. When working with graphs, we identify sets of edges with binary vectors, namely given a graph  $G = (V, E)$ , we identify a subset of edges  $E' \subset E$  with a binary vector  $x = \{0, 1\}^{|E|}$  where  $x_e = 1$  if  $e \in E'$  and  $x_e = 0$  otherwise.

These sets all correspond to "easy" combinatorial problems: for most of them, it is possible to optimize exactly a linear function in polynomial time. The exception is the knapsack, which can be approximated in polynomial time. They also have a practical significance. Source-destination paths correspond to the online shortest-path-routing problem;  $m$ -sets model ad-display optimization; matchings can be used to perform resource allocation in wireless networks.

**$m$ -sets.** The set of  $m$  sets is the set of vectors  $x = \{0, 1\}^d$  which have at most  $m$  non-null entries, namely:

$$\mathcal{X} = \{x \in \{0, 1\}^d : (1, \dots, 1)^\top x \leq m\}.$$

**Spanning trees.** Consider a graph  $G = (V, E)$ . A spanning tree  $x \in \{0, 1\}^{|E|}$  is a subset of edges which covers each vertex  $v \in V$  at least once and forms a tree. The dimension here is the number of edges:  $d = |E|$ .

$$\mathcal{X} = \{x \in \{0, 1\}^{|E|} : \sum_{e: v \in e} x_e \geq 1 \forall v \text{ and } x \text{ is a tree}\}.$$

**Matroids.** A matroid  $\mathcal{I}$  over a set with  $d$  elements is a set of vectors  $x \in \{0, 1\}^d$  which verify two properties: (i) the inclusion property: if  $x \leq x'$  and  $x' \in \mathcal{I}$  then  $x \in \mathcal{I}$  and (ii) the exchange property: if  $x \in \mathcal{I}$ ,  $x' \in \mathcal{I}$  and  $\mathbf{1}^\top x < \mathbf{1}^\top x'$ , then there exists an index  $i$  such that  $x_i = 0$ ,  $x'_i = 1$ , and  $x + e_i \in \mathcal{I}$ . Since  $m$ -sets and spanning trees of a graph form a matroid, any algorithm for matroids also applies to  $m$ -sets and spanning trees.

**Source-destination paths.** Consider  $G = (V, E)$  a directed acyclic graph and  $u$  and  $v$  in  $V$  two vertices. The set of paths between source  $u$  and destination  $v$  is defined as:

$$\mathcal{X} = \{x \in \{0, 1\}^{|E|} : \sum_{e \in \text{in}(w)} x_e - \sum_{e \in \text{out}(w)} x_e = \mathbf{1}\{w = u\} - \mathbf{1}\{w = v\}, w \in V\}.$$

where  $\text{in}(v)$  and  $\text{out}(v)$  are the set of incoming and outgoing edges respectively of  $v \in V$ . The dimension  $d = |E|$  is the number of edges and  $m = \max_{x \in \mathcal{X}} \mathbf{1}^\top x$  is the length of the longest path from  $u$  to  $v$ .

**Matchings.** Consider  $G = (V, E)$  a bipartite graph. A matching is a set of edges which cover each vertex  $v \in V$  at most once, and the set of matchings is:

$$\mathcal{X} = \{x \in \{0, 1\}^{|E|} : \sum_{e \in E: v \in e} x_e \leq 1 \forall v\}.$$

**Intersection of two matroids.** Consider  $\mathcal{I}$  and  $\mathcal{I}'$  two matroids over a common set with  $d$  elements, their intersection is:

$$\mathcal{X} = \{x \in \{0, 1\}^d : x \in \mathcal{I}, x \in \mathcal{I}'\}.$$

Since the set of matchings of any bipartite graph is the intersection of two matroids, algorithms for the intersection of two matroids also apply to matchings of a bipartite graph.

**Knapsack-like sets.** Consider  $A \in \mathbb{N}^{d \times k}$  matrix and  $c \in \mathbb{N}^k$  a vector both with positive integer entries. The corresponding knapsack set is:

$$\mathcal{X} = \{x \in \{0, 1\}^d : Ax \leq c\}.$$

We call  $\mathcal{X}$  a knapsack-like set as its elements  $x$  verify  $k$  knapsack constraints  $\sum_{i=1}^d A_{\ell,i} x_i \leq c_\ell$ ,  $\ell = 1, \dots, k$ . For  $k = 1$ ,  $\mathcal{X}$  is the set of feasible solutions of a knapsack problem and the set of  $m$ -sets is a knapsack-like set.

## 2.3 Maximization Problems

In order to select a decision  $x(t)$  at time  $t$ , most known algorithms involve maximizing some function over the set of decisions  $\mathcal{X}$ . Hence the computational complexity of these algorithms depends mostly on the complexity of these optimization problems. We consider  $a, b$  two vectors of  $\mathbb{R}^d$  with positive entries.

**Linear Maximization.** Problem  $(P_1)$  involves maximizing a linear function over  $\mathcal{X}$ :

$$\max_x \{a^\top x\} \text{ subject to } x \in \mathcal{X} \quad (P_1)$$

**Index Maximization.** Problem  $(P_2)$  involves maximizing the sum of a linear function and the square root of a linear function over  $\mathcal{X}$ :

$$\max_x \{a^\top x + \sqrt{b^\top x}\} \text{ subject to } x \in \mathcal{X} \quad (P_2)$$

**Budgeted Linear Maximization.** Problem  $(P_3)$  involves maximizing a linear function over  $\mathcal{X}$  subject to a linear budget constraint:

$$\max_x \{b^\top x\} \text{ subject to } x \in \mathcal{X} \text{ and } a^\top x \geq s \quad (P_3)$$

Table 1: Regret and complexity of algorithms.

| ALGORITHM | BEST REGRET BOUND                                                 | COMPLEXITY                                    |
|-----------|-------------------------------------------------------------------|-----------------------------------------------|
| CUCB      | $\mathcal{O}\left(\frac{dm(\ln T)}{\Delta_{\min}}\right)$         | $\mathcal{O}(\mathbf{poly}(d))$               |
| TS        | $\mathcal{O}\left(\frac{dm(\ln T)}{\Delta_{\min}}\right)$         | $\mathcal{O}(\mathbf{poly}(d))$               |
| ESCB      | $\mathcal{O}\left(\frac{d(\ln m)^2(\ln T)}{\Delta_{\min}}\right)$ | $\mathcal{O}( \mathcal{X} )$                  |
| AESCB     | $\mathcal{O}\left(\frac{d(\ln m)^2(\ln T)}{\Delta_{\min}}\right)$ | $\mathcal{O}(\delta_t^{-1} \mathbf{poly}(d))$ |

### 3 Related Work and Contribution

The study of classical bandits dates back to [24] and [17]. The order optimal regret in this problem is  $R(T) = \mathcal{O}\left(\frac{d(\ln T)}{\Delta_{\min}}\right)$ , which is attained by algorithms such as UCB1 [4] and KL-UCB [7]. Linear bandits extend classical bandits when the expected reward is a linear function of the decision [13]. When the set of decisions is a combinatorial set, we have a combinatorial bandit which comes in two version: full-bandit feedback and semi-bandit feedback [8].

We consider combinatorial semi-bandits with independent rewards across items. For such problems, the first lower bound is due to [15] and is specific to matroid bandits, a very specific kind of combinatorial bandit. The best known regret bounds are  $R(T) = \mathcal{O}\left(\frac{dm(\ln T)}{\Delta_{\min}}\right)$  for CUCB [9, 16] and Thompson sampling [27, 26]<sup>1</sup>. Both these algorithms enjoy a polynomial time complexity, as they only solve optimization problem  $(P_1)$ . The ESCB algorithm was proposed in [12] where the authors prove a  $R(T) = \mathcal{O}\left(\frac{d\sqrt{m}(\ln T)}{\Delta_{\min}}\right)$  regret bound. As a follow-up, [14] propose OLS-UCB which, in our problem, reduces to ESCB, and thereby prove that ESCB in fact achieves  $R(T) = \mathcal{O}\left(\frac{d(\ln m)^2(\ln T)}{\Delta_{\min}}\right)$ . Therefore, ESCB achieves the best known regret bound for the problem at hand; however, its computational complexity is not polynomial in the dimension. Namely, ESCB must, at each step, solve optimization problem  $(P_2)$ , which is in general  $\mathcal{NP}$ -hard [3]. For  $m$ -sets, paths, and matchings,  $P_2$  is  $\mathcal{NP}$ -hard, see Section A of the appendix for a more in-depth discussion. There seems to be an interesting interplay between statistical efficiency (regret) and computational complexity, which is summarized in Table 1.

We also highlight that there exist algorithms for particular combinatorial semi-bandits (for  $m$ -sets, spanning trees, and more generally matroids) with both polynomial complexity and order optimal regret  $R(T) = \mathcal{O}\left(\frac{d(\ln T)}{\Delta_{\min}}\right)$ , see [2, 25, 21]. However, those algorithms do not extend to general combinatorial bandits, unlike ESCB, and to some extent AESCB.

Much less is known about combinatorial bandits with full-bandit feedback, which is a harder problem. Of course, combinatorial bandits with full-bandit feedback are a particular case of linear bandits, so that algorithms from [13, 1, 10] can be applied. The state of the art is [23], which proposes an algorithm with regret  $R(T) = \mathcal{O}\left(\frac{dm(\ln T)}{\Delta_{\min}}\right)$ . However, this algorithm only works for  $m$ -sets, and whether or not the same regret can be achieved for any combinatorial set  $\mathcal{X}$  is an open problem.

**Our Main Contribution.** We propose AESCB (approximate ESCB), the first algorithm that is both computationally and statistically efficient for this problem: it has a state-of-the-art regret bound and a polynomial time complexity. We show that, whenever budgeted linear maximization over  $\mathcal{X}$  can be solved up to a given approximation ratio, AESCB is implementable in polynomial time. More precisely, its regret is  $R(T) = \mathcal{O}\left(\frac{d(\ln m)^2(\ln T)}{\Delta_{\min}}\right)$  and its complexity is  $\mathcal{O}(\mathbf{poly}(d, \delta_T^{-1}))$  for a large class of combinatorial semi-bandits, where  $\delta_t$  is any function which vanishes as  $t \rightarrow \infty$ , for instance  $\delta_t = (\ln t)^{-1}$  or  $\delta_t = (\ln \ln t)^{-1}$ . We release the code (in Julia [6]) of AESCB for others to experiment with it.

<sup>1</sup>The authors of [26] study a slightly more general problem; in appendix, we show that their regret bound reduces to  $R(T) = \mathcal{O}\left(\frac{dm(\ln T)}{\Delta_{\min}}\right)$  in the problem we study.

Table 2: Complexity of algorithms (at time step  $t$ ) as a function of the chosen discretisation  $\delta_t$ .

| ALGORITHM      | CUCB & TS              | AESCB                                    |
|----------------|------------------------|------------------------------------------|
| $m$ -SETS      | $\mathcal{O}(d \ln d)$ | $\mathcal{O}(dm\delta_t^{-2})$           |
| SPANNING TREES | $\mathcal{O}(d \ln d)$ | $\mathcal{O}(md(\ln d)^3 \delta_t^{-1})$ |
| MATROIDS       | $\mathcal{O}(d \ln d)$ | $\mathcal{O}(md(\ln d)^3 \delta_t^{-1})$ |
| PATHS          | $\mathcal{O}(d \ln d)$ | $\mathcal{O}(d^2 + dm\delta_t^{-1})$     |
| MATCHINGS      | $\mathcal{O}(m^3)$     | $\mathcal{O}(md^{10}\delta_t^{-1})$      |

## 4 Exact and Approximate ESCB Algorithms

We propose the AESCB algorithm for combinatorial semi-bandits, which is an approximate version of the ESCB algorithm. We prove that AESCB and ESCB both enjoy the same regret bound. In the subsequent sections, we will also show that AESCB, unlike ESCB, can be implemented in polynomial time with respect to the dimension for a large class of combinatorial problems.

We define the following statistics, for  $i = 1, \dots, d$ :

$$\begin{aligned}
n_i(t) &= \sum_{t'=1}^{t-1} x_i(t') \\
\hat{\theta}(t) &= \frac{\sum_{t'=1}^{t-1} x_i(t') Z_i(t')}{\max(1, \sum_{t'=1}^{t-1} x_i(t'))} \\
\sigma_i^2(t) &= \begin{cases} \frac{f(t)}{2n_i(t)} & \text{if } n_i(t) \geq 1 \\ +\infty & \text{otherwise.} \end{cases}
\end{aligned}$$

where, at time  $t$ ,  $n_i(t)$  is the number of samples obtained for  $\theta_i$ ,  $\hat{\theta}_i(t)$  is the estimate of  $\theta_i$ , and  $\sigma_i^2(t)$  is proportional to the variance of estimate  $\hat{\theta}_i(t)$ .  $f(t)$  is defined as  $\ln t + 4m \ln \ln t$ . We denote by  $n(t) = (n_i(t))_{i=1, \dots, d}$ ,  $\hat{\theta}(t) = (\hat{\theta}_i(t))_{i=1, \dots, d}$ , and  $\sigma^2(t) = (\sigma_i^2(t))_{i=1, \dots, d}$  the corresponding vectors.

### 4.1 The ESCB Algorithm

**Definition 4.1 (ESCB)** *The ESCB algorithm is the policy which at any time  $t \geq 1$  selects decision:*

$$x(t) \in \arg \max_{x \in \mathcal{X}} \{ \hat{\theta}(t)^\top x + \sqrt{\sigma^2(t)^\top x} \}$$

where ties are broken arbitrarily.

The ESCB algorithm is an optimistic algorithm, where the index  $\hat{\theta}(t)^\top x + \sqrt{\sigma^2(t)^\top x}$  serves as an upper confidence bound of the unknown reward of decision  $x$ ,  $\theta^\top x$ . The nonlinear term is the estimated standard deviation of the reward, as the random variables are considered to be independent. Also, ESCB is a natural extension of the UCB1 algorithm for classical bandits. In order to implement ESCB, one needs to solve the optimization problem  $(P_2)$  at each time step.

The regret of ESCB was analyzed by [12] and then improved by [14]. The regret bound is presented in Theorem 4.2, as the algorithm of [14] reduces to that of [12] when rewards are uncorrelated across items.

**Theorem 4.2 (Regret of ESCB)** *The regret of ESCB admits the following upper bound for all  $T \geq 1$ :*

$$R(T) \leq C_4(m) + \frac{2dm^3}{\Delta_{\min}^2} + \frac{96df(T)}{\Delta_{\min}} \left\lceil \frac{\ln m}{1.61} \right\rceil^2$$

with  $f(t) = \ln t + 4m \ln \ln t$  and  $C_4(m)$  a positive number that solely depends on  $m$ .

By corollary:

$$R(T) = \mathcal{O}\left(d(\ln m)^2 \frac{1}{\Delta_{\min}} \ln T\right) \quad \text{as } T \rightarrow \infty.$$

The regret upper bound of Theorem 4.2 is the best known regret upper bound for combinatorial semi-bandits with independent rewards across items, so that ESCB is, to the best of our knowledge, the state-of-the-art algorithm for this problem in terms of regret. However, ESCB involves solving optimization problem  $(P_2)$  at each step, and this problem is  $\mathcal{NP}$ -hard [3], so one cannot implement it efficiently as is.

## 4.2 AESCB

We now propose AESCB (Approximate-ESCB), an algorithm that approximates ESCB and enjoys the same regret bound, while being implementable with polynomial complexity. The AESCB algorithm requires two sequences  $(\varepsilon_t, \delta_t)$ , which quantify the level of approximation at each time step.

**Definition 4.3 (AESCB)** *The AESCB algorithm with approximation factors  $(\varepsilon_t, \delta_t)_{t \geq 1}$  is the policy which at any time  $t \geq 1$  selects a decision  $x(t)$  verifying:*

$$\arg \max_{x \in \mathcal{X}} \{ \hat{\theta}(t)^\top x + \sqrt{\sigma^2(t)^\top x} \} \leq \delta_t + \hat{\theta}(t)^\top x(t) + \frac{1}{\varepsilon_t} \sqrt{\sigma^2(t)^\top x(t)}$$

where ties are broken arbitrarily.

When  $(\varepsilon_t, \delta_t) = (1, 0)$  for all  $t \geq 1$ , AESCB reduces to ESCB. The rationale is that ESCB requires to solve optimization problem  $(P_2)$  at each time step, and while  $(P_2)$  is  $\mathcal{NP}$ -hard and cannot be solved exactly in polynomial time (unless  $\mathcal{P} = \mathcal{NP}$ ), it can be approximated in polynomial time in many cases of interest, so that AESCB lends itself to polynomial-time implementation. We show how to do this in Section 5.

## 4.3 Regret Analysis of AESCB

Our first main result is Theorem 4.4, which provides a regret upper bound for AESCB. We show that, if one chooses approximation parameters  $(\varepsilon_t, \delta_t)$  with  $\varepsilon_t = \varepsilon > 0$  some fixed number and  $\delta_t$  any sequence such that  $\lim_{t \rightarrow \infty} \delta_t = 0$ , then AESCB verifies the same (state-of-the-art) regret as ESCB up to a multiplicative constant. For  $m$ -sets, knapsack sets, and source destination paths, we choose  $\varepsilon = 1$ . For spanning trees, matroids, matchings, and matroid intersection, we choose  $\varepsilon = \frac{1}{2}$  (see Section 5). This choice of parameters does not require any knowledge about the time horizon  $T$ , nor about the unknown problem parameters  $\theta$ , nor about the minimal gap  $\Delta_{\min}$ . Nevertheless, if  $\Delta_{\min}$  is known as well, we can select  $\delta_t$  to yield an even better algorithm; however, knowing this parameter is by no means required.

We will show that, with this choice of parameters, AESCB can be implemented in polynomial time (see Section 5). A sketch of proof for Theorem 4.4 is presented in the next subsection to further highlight the algorithm rationale, while the complete proof is presented in appendix.

**Theorem 4.4 (Regret of AESCB)** *The regret of AESCB with parameters  $(\varepsilon_t, \delta_t)$  admits the following upper bound for all  $T \geq 1$ :*

$$R(T) \leq C_4(m) + \frac{2dm^3}{\Delta_{\min}^2} + \frac{24df(T)}{(\min_{t \leq T} \varepsilon_t)^2 \Delta_{\min}} \left\lceil \frac{\ln m}{1.61} \right\rceil^2 + 4 \sum_{t=1}^T \delta_t \mathbf{1}(\Delta_{\min} \leq 4\delta_t).$$

with  $f(t) = \ln t + 4m \ln \ln t$  and  $C_4(m)$  a positive number that solely depends on  $m$ .

By corollary, for  $\varepsilon_t = \varepsilon$  and  $\lim_{t \rightarrow \infty} \delta_t = 0$ , we have:

$$R(T) = \mathcal{O} \left( d (\ln m)^2 \frac{1}{\Delta_{\min}} \ln T \right) \quad \text{as } T \rightarrow \infty.$$

Similarly, with  $\varepsilon_t = \varepsilon$  and  $\delta_t < \frac{1}{4} \Delta_{\min}$ , we have, for all  $T \geq 1$ :

$$R(T) \leq C_4(m) + \frac{2dm^3}{\Delta_{\min}^2} + \frac{24df(T)}{\varepsilon^2 \Delta_{\min}} \left\lceil \frac{\ln m}{1.61} \right\rceil^2.$$

#### 4.4 Theorem 4.4: Sketch of Proof

The regret analysis of AESCB involves upper bounding the reward gap of the decision chosen at time  $t$ ,  $\Delta_{x(t)}$ , by considering three cases. Define the following events:

$$\begin{aligned}\mathcal{A}_t &= \left\{ \exists x \in \mathcal{X} : |(\theta - \hat{\theta}(t))^\top x| \geq \sqrt{\sigma^2(t)^\top x} \right\} \\ \mathcal{B}_t &= \{\Delta_{x(t)} \leq 4\delta_t\}\end{aligned}$$

If  $\mathcal{A}_t$  occurs, since  $\theta \in [0, 1]^d$  and  $\mathbf{1}^\top x^\star \leq m$ :

$$\Delta_{x(t)} \leq \theta^\top x^\star \leq m.$$

If  $\mathcal{B}_t$  occurs, by definition:

$$\Delta_{\min} \leq \Delta_{x(t)} \leq 4\delta_t.$$

If  $\mathcal{C}_t = \overline{\mathcal{A}_t} \cup \overline{\mathcal{B}_t}$  occurs, the index of the optimal decision is greater than the optimal reward  $\theta^\top x^\star$ :

$$\begin{aligned}\theta^\top x^\star &\leq \hat{\theta}(t)^\top x^\star + \sqrt{\sigma^2(t)^\top x^\star} \\ &\leq \delta_t + \hat{\theta}(t)^\top x(t) + \frac{1}{\varepsilon_t} \sqrt{\sigma^2(t)^\top x(t)} \\ &\leq \delta_t + \theta^\top x(t) + \frac{2}{\varepsilon_t} \sqrt{\sigma^2(t)^\top x(t)} \\ &\leq \frac{1}{4} \Delta_{x(t)} + \theta^\top x(t) + \frac{2}{\varepsilon_t} \sqrt{\sigma^2(t)^\top x(t)}.\end{aligned}$$

where we used the definition of AESCB, and the fact that  $\mathcal{A}_t$  and  $\mathcal{B}_t$  do not occur. Therefore, if  $\mathcal{C}_t$  occurs:

$$\Delta_{x(t)} \leq \frac{8}{3\varepsilon_t} \sqrt{\sigma^2(t)^\top x(t)} \leq \frac{4}{\varepsilon_t} \sqrt{\sigma^2(t)^\top x(t)}.$$

Putting it together, we get:

$$\begin{aligned}\Delta_{x(t)} &\leq \Delta_{x(t)} \left( \mathbf{1}(\mathcal{A}_t) + \mathbf{1}(\mathcal{B}_t) + \mathbf{1}(\mathcal{C}_t) \right) \\ &\leq m \mathbf{1}(\mathcal{A}_t) + 4\delta_t \mathbf{1}(\Delta_{\min} \leq 4\delta_t) + \Delta_{x(t)} \mathbf{1} \left\{ \Delta_{x(t)} \leq \frac{4}{\varepsilon_t} \sqrt{x(t)^\top \sigma^2(t)} \right\}.\end{aligned}$$

Taking expectations and summing over  $t$ :

$$\begin{aligned}R(T) &= \sum_{t=1}^T \mathbb{E}(\Delta_{x(t)}) \\ &\leq \sum_{t=1}^T m \mathbb{P}(\mathcal{A}_t) + \sum_{t=1}^T 4\delta_t \mathbf{1}(\Delta_{\min} \leq 4\delta_t) + \sum_{t=1}^T \mathbb{E} \left( \Delta_{x(t)} \mathbf{1} \left\{ \Delta_{x(t)} \leq \frac{4}{\varepsilon_t} \sqrt{x(t)^\top \sigma^2(t)} \right\} \right).\end{aligned}$$

The first term is bounded by a constant, since, using a concentration inequality, we may show that  $\mathcal{A}_t$  occurs with small probability. The last term can be bounded using (rather intricate) counting arguments as in the analysis of ESCB. The complete proof is presented in appendix.

## 5 AESCB in Polynomial Time

We now show a technique to implement AESCB that ensures polynomial time complexity. While our methodology is generic, the precise value of the computational complexity depends on the combinatorial set  $\mathcal{X}$ , and will be explained in details in Sections 5.1 – 5.5. Our approach involves three steps: rounding and scaling to ensure that the weights are integer, then solving the budgeted linear maximization ( $P_3$ ) several times, and finally maximizing over the budget to obtain the result. Given time  $t$ , statistics  $\hat{\theta}(t)$  and  $\sigma^2(t)$ , and approximation factors  $(\varepsilon_t, \delta_t)$ , the method works as follows.

**Step 1: rounding and scaling.** Define  $a(t)$  and  $b(t)$ :

$$\begin{aligned}\xi(t) &= \lceil m/\delta_t \rceil. \\ a_i(t) &= \lceil \xi(t) \hat{\theta}_i(n) \rceil, \quad i \in \{1, \dots, d\} \\ b_i(t) &= \xi(t)^2 \sigma_i^2(t), \quad i \in \{1, \dots, d\}\end{aligned}$$

**Step 2: budgeted linear maximization.** For all  $s \in \{0, \dots, m\xi(t)\}$ , compute  $\bar{x}^s(t)$ , an  $\varepsilon_t$ -optimal solution to budgeted linear maximization problem  $(P_3)$ :

$$\bar{x}^s(t) \geq \varepsilon_t \left( \max_{x \in \mathcal{X}: a(t)^\top x \geq s} \{b(t)^\top x\} \right) \quad \text{and} \quad a(t)^\top \bar{x}^s(t) \geq s.$$

**Step 3: optimizing over a budget.** Return decision  $x(t)$ :

$$\begin{aligned}x(t) &= \bar{x}^{s^*(t)}(t) \text{ with} \\ s^*(t) &\in \arg \max_{s=0, \dots, m\xi(t)} \left\{ s + \frac{1}{\varepsilon_t} \sqrt{b(t)^\top \bar{x}^s(t)} \right\}.\end{aligned}$$

$a(t)$  is defined using a ceiling operation in order to ensure that  $a(t)^\top x$  has an integer value for any  $x \in \mathcal{X}$ , while  $b(t)$  does not need to have integer entries. Theorem 5.1 (see proof in appendix) states that this technique returns the decision chosen by AESCB, in a time proportional to solving the optimization problem  $(P_3)$  at most  $m\xi(t)$  times (where  $\xi(t)$  is bounded by a polynomial in  $d$ ), and that the input parameters  $a(t)$  and  $b(t)$  of  $(P_3)$  are positive vectors and where the entries of  $a(t)$  are in  $\{1, \dots, \xi(t)\}$ . In the next subsections, we show that, for many combinatorial sets of interest, one can solve  $(P_3)$  with a time complexity bounded by a polynomial in the dimension, which implies that AESCB is indeed implementable in polynomial time.

**Theorem 5.1** *The above algorithm returns a decision  $x(t) \in \mathcal{X}$  verifying the AESCB definition:*

$$\arg \max_{x \in \mathcal{X}} \{ \hat{\theta}(t)^\top x + \sqrt{\sigma^2(t)^\top x} \} \leq \delta_t + \hat{\theta}(t)^\top x(t) + \frac{1}{\varepsilon_t} \sqrt{\sigma^2(t)^\top x}$$

*It does so by solving optimization problem  $(P_3)$  at most  $m\xi(t)$  times with input parameters  $a(t)$  and  $b(t)$ , where  $a(t) \in \{1, \dots, \xi(t)\}^d$  and  $b(t) \in \mathbb{R}^d$ .*

Based on Theorem 5.1, we now highlight how to find  $\varepsilon$ -optimal solutions to the optimization  $(P_3)$ . The complexity at each time step of AESCB with parameters  $(\varepsilon_t, \delta_t)$  is summarized in Table 2, where  $\xi(t) = \lceil m\delta_t^{-1} \rceil$ .

Since  $\delta_t$  can be chosen as a function that vanishes arbitrarily slowly, the dependency of the complexity on  $t$  in Table 2 can be made arbitrarily mild. Additionally, if  $\Delta_{\min}$  is known to the decision maker, one can set  $\delta_t = \frac{1}{4}\Delta_{\min}$ , so that the computational complexity does not even depend on  $t$  altogether. We emphasize once again that knowing  $\Delta_{\min}$  is by no means necessary.

We now consider time  $t$  fixed and we drop the time index to simplify notation. We consider input parameters  $a$  and  $b$  for  $(P_3)$  with  $a \in \{1, \dots, \xi\}$ . We do so for each type of combinatorial set in Sections 5.1 – 5.5. We provide a description of the algorithm, with pseudocode given in appendix.

## 5.1 $m$ -sets

**Claim 5.2** *Optimization problem  $(P_3)$  with  $\mathcal{X}$  the set of  $m$ -sets can be solved exactly (i.e.  $\varepsilon = 1$ ) in time  $\mathcal{O}(m^2 d \xi)$  using the algorithm below.*

In fact, since  $m$ -sets are a particular cases of a knapsack set with matrix  $A = (1, \dots, 1)^\top$ ,  $k = 1$  and  $c = (m)$ , we can simply apply the algorithm for knapsack sets explained in Section 5.2 below.

## 5.2 Knapsack sets

**Claim 5.3** *Optimization problem  $(P_3)$  with  $\mathcal{X}$  a knapsack set can be solved exactly (i.e.  $\varepsilon = 1$ ) in time  $\mathcal{O}((\prod_{\ell=1}^k c_\ell)dm\xi)$  using the algorithm below.*

For  $i \in \{0, \dots, d\}$ , define the optimization problem

$$\begin{aligned} & \max_x b^\top x \\ & \text{subject to } Ax \leq c, \quad a^\top x \geq s, \\ & \sum_{j=1}^i x_j = 0, \quad x \in \{0, 1\}^d \quad (P_4(s, c, i)) \end{aligned}$$

where we recall that  $c \in \mathbb{N}^k$  is a vector with integer entries. We denote by  $\mathcal{V}_4(s, c, i)$  the optimal value of this optimization program. Since  $(P_4(s, c, 0))$  reduces to  $(P_3)$ , it is sufficient to solve  $(P_4(s, c, i))$  for  $i \in \{0, \dots, d\}$ . We do so using dynamic programming. Let  $x^*$  an optimal solution to  $(P_4(s, c, i))$ . If  $x_{i+1}^* = 1$ , then

$$A(x^* - e_i) = Ax^* - Ae_i \leq c - Ae_i$$

hence  $x^* - e_i$  is an optimal solution to  $(P_4(\max(s - a_i, 0), c - Ae_i, i + 1))$ . If  $x_i^* = 0$ , then  $x^*$  is an optimal solution to  $(P_4(s, c, i + 1))$ . Therefore,

$$\mathcal{V}_4(s, c, i) = \max\{\mathcal{V}_4(s, c, i + 1), a_i + \mathcal{V}_4(\max(s - a_i, 0), c - Ae_i, i + 1)\}$$

By recursion over  $i$ ,  $c$ , and  $s$ , we can compute the value  $\mathcal{V}_4(s, c', i)$  for  $s \in \{0, \dots, m\xi\}$ ,  $c' \in \mathbb{N}^k$  with  $c' \leq c$ , and  $i \in \{0, \dots, d\}$  in time  $\mathcal{O}((\prod_{\ell=1}^k c_\ell)dm\xi)$ . The solution to  $(P_3)$ , denoted by  $x^*$ , is then:

$$x_i^* = \begin{cases} 0 & \text{if } \mathcal{V}_4(s, c, i) = \mathcal{V}_4(s, c, i + 1) \\ 1 & \text{otherwise.} \end{cases}$$

Therefore, we can solve  $(P_3)$  for all  $s \leq m\xi$  in time  $\mathcal{O}((\prod_{\ell=1}^k c_\ell)dm\xi)$ . In particular, for  $m$ -sets, we can do so with a time complexity of  $\mathcal{O}(dm^2\xi)$ .

## 5.3 Source destination paths

**Claim 5.4** *Optimization problem  $(P_3)$  with  $\mathcal{X}$  the set of paths between source  $u$  and destination  $v$  in  $G = (V, E)$  a directed acyclic graph can be solved exactly (i.e.  $\varepsilon = 1$ ) in time  $\mathcal{O}(m\xi|E| + |V| \ln |V|)$  using algorithm below.*

Consider  $v$  fixed throughout, and denote by  $(P_3(u, s))$  this optimization problem and  $\mathcal{V}_3(u, s)$  its optimal value. If  $s \leq 0$ ,  $(P_3(u, s))$  is simply the problem of finding the path from  $u$  to  $v$  maximizing  $b^\top x$ , since  $a$  has positive entries so that  $a^\top x \geq 0 \geq s$ , for all  $x \in \mathcal{X}$ . Hence, we can compute  $(P_3(u, s))$  for all  $u$  with the Bellman-Ford algorithm in time  $\mathcal{O}(|V||E|)$ .

If  $s > 0$ , let  $x^*$  an optimal solution to  $(P_3(u, s))$ . Since  $x^*$  is a path from  $u$  to  $v$ , there exists a unique  $w \in V$  such that  $x_{(u,w)}^* = 1$ , and  $(x^* - e_{(u,w)})$  is a path from  $w$  to  $v$ . In turn, we must have that  $(x^* - e_{(u,w)})$  is an optimal solution to  $(P_3(w, \max(s - a_{(u,w)}, 0)))$ . Therefore, we have the following dynamic programming equation:

$$\mathcal{V}_3(u, s) = \max_{w: (u,w) \in E} \{b_{(u,w)} + \mathcal{V}_3(w, \max\{s - a_{(u,w)}, 0\})\}$$

As  $a \in \{1, \dots, \xi\}^d$ , if  $\mathcal{V}_3(u, s')$  is known for all  $u \in V$  and all  $s' \in \{0, \dots, s - 1\}$ , then applying the above relationship enables us to compute  $\mathcal{V}_3(u, s)$  for all  $u \in V$ . By recursion, we can compute  $\mathcal{V}_3(u, s)$  for all  $s \in \{0, \dots, m\xi\}$  in time  $\mathcal{O}((m\xi + |V|)|E|)$ . The solution to  $(P_3)$ , denoted by  $x^*$ , can also be computed by recursion. Using the same dynamic programming principle, denote by  $x^*(u, s)$  the solution of  $(P_3(u, s))$ , we have:

$$\begin{aligned} x^*(u, s) &= e_{(u, w^*(u, s))} + x^*(w^*(u, s), s - a_{(u, w^*(u, s))}) \text{ with} \\ w^*(u, s) &\in \arg \max_{w: (u,w) \in E} \{b_{(u,w)} + \mathcal{V}_3(w, \max(s - a_{(u,w)}, 0))\} \end{aligned}$$

By recursion, we can compute the solution to  $(P_3)$  for all  $s \in \{0, \dots, m\xi\}$  in time  $\mathcal{O}((m\xi + |V|)|E|)$ .

## 5.4 Spanning Trees and Matroids

**Claim 5.5** *Optimization problem  $(P_3)$  with  $\mathcal{X}$  the set of spanning trees of graph  $G = (V, E)$  can be solved with approximation ratio  $\varepsilon = \frac{1}{2}$  in time  $\mathcal{O}(|E|(\ln |V|)^2 + |V|(\ln |E|)^3)$  using the algorithm below. The same holds for any matroid.*

The algorithm is made of four steps, and is similar to that of [22] (see this reference for further details).

**Step 1: Lagrangian relaxation.** De

$$\max_x \{b^\top x + \lambda(a^\top x - s)\} \text{ subject to } x \in X$$

Denote by  $M(\lambda)$  its value and  $\mathcal{L}(\lambda)$  the set of optimal solutions. Define  $\lambda^* = \arg \min_{\lambda \geq 0} M(\lambda)$ . Computing  $M(\lambda)$  can be done in polynomial time using a greedy algorithm, since it is equivalent to maximizing a linear function over a matroid [20]. Furthermore,  $\lambda^*$  can be found using Meggido's search technique [19], as it involves minimizing a piecewise linear function.

**Step 2: candidate solutions.** For an arbitrarily small  $\varepsilon > 0$ , if  $|\lambda - \lambda^*| < \varepsilon$ , we must have that  $\mathcal{L}(\lambda) \subset \mathcal{L}(\lambda^*)$ . Therefore, by solving the Lagrangian relaxation of the problem for  $\lambda = \lambda^* + \varepsilon$  and  $\lambda = \lambda^* - \varepsilon$ , we obtain two solutions  $x^+$  and  $x^-$  in  $\mathcal{L}(\lambda^*)$  with  $a^\top x^+ \geq s$  and  $a^\top x^- \leq s$ .

**Step 3: solution refining.** We now use an iterative procedure in order to find a good solution using candidates  $x^+$  and  $x^-$ . Consider  $e, e' \in E$  such that  $x_e^+ = x_{e'}^- = 1$  and  $x_{e'}^+ = x_e^- = 0$  and define  $x = x^+ \setminus \{e\} \cup \{e'\}$ . If  $a^\top x \geq s$ , then replace  $x^+$  by  $x$  and otherwise replace  $x^-$  by  $x$  and repeat this procedure until  $x^+$  and  $x^-$  differ by exactly one element. Finally, return  $x^+$ . At each step of this procedure:

$$(b + \lambda a)^\top x = (b + \lambda a)^\top x^+ = (b + \lambda a)^\top x^-$$

therefore  $x \in \mathcal{L}(\lambda^*)$ . Denote by  $x^*$  the solution of  $(P_3)$ .

Since  $x^+$  and  $x^-$  are in  $\mathcal{L}(\lambda^*)$ :

$$b^\top x^- + \lambda(a^\top x^- - s) \geq b^\top x^* + \lambda(a^\top x^* - s)$$

Since  $a^\top x^- \leq s \leq a^\top x^*$ , we deduce that  $b^\top x^- \geq b^\top x^*$ . Since  $x^+$  and  $x^-$  differ by at most one element:

$$b^\top x^+ \geq b^\top x^- - \max_{e \in E} b_e \geq b^\top x^* - \max_{e \in E} b_e$$

So, after steps 1-3, we get  $x$  such that  $a^\top x \geq s$  and  $b^\top x \geq b^\top x^* - \max_{e \in E} b_e$ .

**Step 4: a  $\frac{1}{2}$  optimal solution.** Lastly, we search over the two edges with largest weight to obtain a constant multiplicative approximation factor. For all sets of two edges  $E'' \subset E$ ,  $|E''| = 2$ , define  $G' = (V, E')$  where

$$E' = \{e \in E \setminus E'' : b_e \leq \min_{e'' \in E''} \{b_{e''}\}\}$$

and apply steps 1 to 3 to solve the problem where  $G$  and  $s$  are replaced by  $G'$  and  $s' = s - \sum_{e'' \in E''} a_{e''}$ , where  $x'(E'')$  is the solution found by steps 1-3. Finally, return  $x(E'') = x'(E'') + \sum_{e'' \in E''} e_{e''}$ , for the value of  $E''$  maximizing  $a^\top x(E'')$ . This yields a  $1/2$  optimal solution in time  $\mathcal{O}(|E|^2(\ln |V|)^2 + |V|(\ln |E|)^3)$  by the same arguments as that used in [22, 5].

## 5.5 Matchings and Matroid Intersection

**Claim 5.6** *Optimization problem  $(P_3)$  with  $\mathcal{X}$  the set of matchings of a bipartite graph  $G = (V, E)$  can be solved with approximation ratio  $\varepsilon = \frac{1}{2}$  in time  $\mathcal{O}(|V|^3|E|^4)$  using the algorithm below. The same holds for any intersection of two matroids.*

The algorithm is made of four steps and is very similar of that of [5], which itself is inspired by the algorithm for matroids of [22].

**Step 1: Lagrangian relaxation.** Define the Lagrangian relaxation of the problem:

$$\max_x \{b^\top x + \lambda(a^\top x - s)\} \text{ subject to } x \in X$$

denote by  $M(\lambda)$  its value and  $\mathcal{L}(\lambda)$  the set of optimal solutions. Define  $\lambda^* = \arg \min_{\lambda \geq 0} M(\lambda)$ . Computing  $M(\lambda)$  can be done in time using the Hungarian algorithm, since it is maximizing a linear function over the set of matchings of a bipartite graph. Furthermore,  $\lambda^*$  can be found using Meggido's parametric search technique [19] as it involves minimizing a piecewise linear function.

**Step 2: candidate solutions.** For an arbitrarily small  $\varepsilon > 0$ , if  $|\lambda - \lambda^*| < \varepsilon$  we must have that  $\mathcal{L}(\lambda) \subset \mathcal{L}(\lambda^*)$ . So, by solving the Lagrangian relaxation of the problem for  $\lambda = \lambda^* + \varepsilon$  and  $\lambda = \lambda^* - \varepsilon$ , we obtain two solutions  $x^+$  and  $x^-$  in  $\mathcal{L}(\lambda^*)$  with  $a^\top x^+ \geq s$  and  $a^\top x^- \leq s$ .

**Step 3: solutions refining.** We now use an iterative procedure in order to find a good solution using candidates  $x^+$  and  $x^-$ . Define their symmetric difference  $x' = x^+ \oplus x^-$ .  $x'$  is made of a disjoint union of paths and cycles. Take  $x''$  as one of such paths or cycles, and define  $x = x^- \oplus x''$ . If  $a^\top x \geq s$ , then replace  $x^+$  by  $x$  and otherwise replace  $x^+$  by  $x$ . Repeat this procedure until  $x^+$  and  $x^-$  differ by at most two elements (the symmetric difference  $x^+ \oplus x^-$  decreases at each step). Finally, return  $x^+$ . At each step of this procedure:

$$(b + \lambda^* a)^\top x = (b + \lambda^* a)^\top x^+ = (b + \lambda^* a)^\top x^-$$

therefore  $x \in \mathcal{L}(\lambda^*)$ . Denote by  $x^*$  the solution of  $(P_3)$ .

Since  $x^+$  and  $x^-$  are in  $\mathcal{L}(\lambda^*)$ :

$$b^\top x^- + \lambda^*(a^\top x^- - s) \geq b^\top x^* + \lambda^*(a^\top x^* - s)$$

Since  $a^\top x^- \leq s \leq a^\top x^*$  we deduce that  $b^\top x^- \geq b^\top x^*$ . Since  $x^+$  and  $x^-$  differ by at most two elements:

$$b^\top x^+ \geq b^\top x^- - 2 \max_{e \in E} b_e \geq b^\top x^* - 2 \max_{e \in E} b_e$$

So, after steps 1-3, we get  $x$  such that  $a^\top x \geq s$  and  $b^\top x \geq b^\top x^* - 2 \max_{e \in E} b_e$ .

**Step 4: an  $\frac{1}{2}$  optimal solution.** Lastly, we search over the four edges with largest weight to obtain a constant multiplicative approximation factor. For all sets of four edges  $E'' \subset E$ ,  $|E''| = 4$ , define  $G' = (V, E')$  where

$$E' = \{e \in E \setminus E'' : b_e \leq \min_{e' \in E''} \{b_{e'}\}\}$$

and apply steps 1 to 3 to solve the problem where  $G$  and  $s$  are replaced by  $G'$  and  $s' = s - \sum_{e'' \in E''} a_{e''}$ , where  $x'(E'')$  is the solution found by steps 1-3. Finally, return  $x(E'') = x'(E'') + \sum_{e'' \in E''} e_{e''}$ , for the value of  $E''$  maximizing  $a^\top x(E'')$ . This yields a  $1/2$  optimal solution in time  $\mathcal{O}(|V|^3 |E|^4)$  by the same arguments as that used in [5].

## 6 Numerical Experiments

We evaluate the performance of TS, CUCB, ESCB, and AESCB through numerical experiments. ESCB is implemented by casting the optimization problem  $(P_2)$  as an ISOCP (Integer Second-Order Cone Programming) and using an ISOCP solver, see appendix for more details. As done in all prior work [7, 12], we simulate ESCB and AESCB using  $f(t) = \ln t$ , neglecting the  $(4m \ln \ln t)$  term, which gives much better performance. This issue is discussed in [12]. Our implementation of the four algorithms is available along with this article. For  $m$ -sets, we choose  $m = \lfloor d/3 \rfloor$  and  $\theta_i = 0.55$  for  $i \leq d/2$  and  $\theta_i = 0.4$  for  $i > d/2$ . For source-destination paths, we consider the graph  $G = (V, E)$ , a complete directed acyclic graph, so that  $(i, j) \in E$  if and only if  $i < j$ . The source is 1, the destination is  $|V|$  and  $\theta_{(i,j)} = 0.4$  for  $(i, j) \neq (1, |V|)$  and  $\theta_{(1, |V|)} = 0.55$ . We have  $d = |V|(|V| - 1)/2$ ,  $m = |V| - 1$  and the optimal path is  $\{(i, i + 1) \text{ for } i \in \{1, 2, \dots, |V|\}\}$ . For spanning trees,

we consider the graph  $G = (V, E)$ , a complete graph, and  $\theta_{(i,j)} = 0.4$  for all  $(i, j)$  with  $i \neq 1$  and  $\theta_{(1,j)} = 0.55$ . We have  $d = |V|(|V| - 1)/2$ ,  $m = |V| - 1$ ; thus, the optimal decision is a star network. For matchings, we consider a complete bipartite graph  $G = (V, E)$  with  $V = V_1 \cup V_2$  and  $|V_1| = |V_2|$ ,  $\theta_{(i,j)} = 0.4$  for all  $(i, j)$ ,  $i \neq j$  and  $\theta_{(i,i)} = 0.55$ . The optimal decision is  $x_{(i,j)}^* = \mathbf{1}\{i = j\}$  and  $d = |V_1||V_2|$ ,  $m = \min(|V_1|, |V_2|)$ . In Tables 3 and 4 the results are presented in the format  $\bar{z} \pm \Delta z$  where  $\bar{z}$  is the empirical mean and  $\Delta z$  is 1.96 times the square root of the ratio between the empirical variance and the number of samples.

**Regret.** In Figure 1, we present the expected regret of algorithms (with 95% confidence intervals) averaged over 10 sample paths. We note that (i) the regret of AESCB is very close to that of ESCB, so that the approximation comes at virtually no cost in terms of regret (ii) ESCB and AESCB outperform CUCB for matchings and spanning trees and CUCB performs better than ESCB and AESCB for paths (iii) TS performs well on average; however, its regret has a lot of variability across sample paths, performing quite badly on some of them. Therefore, it is a “risky” algorithm to use, unlike the others.

**Tuning.** In Table 4 we consider tuned versions of CUCB, ESCB and AESCB:

$$x(t) \in \arg \max_{x \in \mathcal{X}} \left\{ \hat{\theta}(t)^\top x + \sum_{i=1}^d x_i \frac{\alpha \ln t}{\sqrt{n_i(t)}} \right\} \quad (\text{tuned-CUCB})$$

$$x(t) \in \arg \max_{x \in \mathcal{X}} \left\{ \hat{\theta}(t)^\top x + \sqrt{2\alpha\sigma^2(t)^\top x} \right\} \quad (\text{tuned-ESCB})$$

$$\arg \max_{x \in \mathcal{X}} \left\{ \hat{\theta}(t)^\top x + \sqrt{2\alpha\sigma^2(t)^\top x} \right\} \leq \delta_t + \hat{\theta}(t)^\top x(t) + \frac{1}{\varepsilon_t} \sqrt{2\alpha\sigma^2(t)^\top x(t)} \quad (\text{tuned-AESCB})$$

where  $\alpha \geq 0$  is a parameter, and  $\alpha = \frac{1}{2}$  corresponds to the normal versions of CUCB, ESCB and AESCB considered in Figure 1. The regret obtained by choosing the best value of  $\alpha$  is presented in bold. In general we notice that tuning  $\alpha$  does yield some improvement in regret with respect to the standard algorithms (for  $\alpha = \frac{1}{2}$ ), however the improvement comes at the cost of careful tuning. We also notice that performing aggressive tuning by selecting  $\alpha$  very small does not seem to yield good performance in general.

**Computation time.** In Table 3, we present the computation times required to select an arm at time  $t = 1000$  for ESCB, AESCB, CUCB, and TS (with 95% confidence intervals) as a function of the problem dimension  $d$ . We observe that the computation time for AESCB indeed appears to grow slowly in  $d$ , and that the computation times for all algorithms have the same magnitude.

## 7 Conclusion

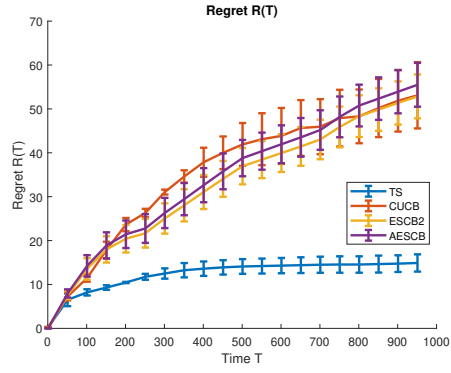
We propose AESCB, the first algorithm which enjoys both the state-of-the art regret bound of ESCB and polynomial computational complexity. We believe our work opens two important research questions: (i) Since TS has generally polynomial complexity and seems to work better than AESCB numerically, can one prove that it also has  $R(T) = \mathcal{O}\left(\frac{d(\ln m)^2(\ln T)}{\Delta_{\min}}\right)$  regret in general? (ii) What is the optimal trade-off between regret and computational complexity in combinatorial bandits?

## A On the NP-hardness of $P_2$

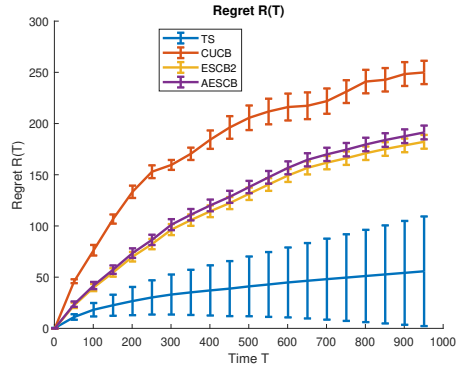
We discuss the  $\mathcal{NP}$ -hardness of  $P_2$  by recalling some results of [3]. In fact the authors prove a more general result which is that  $P_2$  is  $\mathcal{NP}$ -hard when the objective function  $a^\top x + \sqrt{b^\top x}$  is replaced by  $a^\top x + g(b^\top x)$  where  $g$  is any strictly concave function.

**Proposition 1** (Proposition 1 of [3]) Consider  $\mathcal{X} = \{x \in \{0, 1\}^d : \sum_{i=1}^d x_i = m\}$  and  $g$  a strictly concave function. Then maximizing  $x \mapsto a^\top x + g(b^\top x)$  over  $\mathcal{X}$  is  $\mathcal{NP}$ -hard.

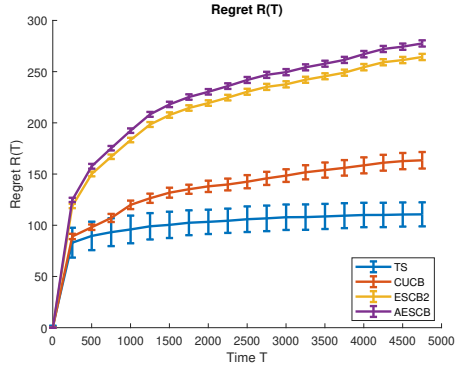
**Proposition 2** (Proposition 3 of [3]) Consider  $G = (V, E)$ , a directed acyclic graph. Consider  $\mathcal{X}$  the set of paths in  $G$  and  $g$  a strictly concave function. Then maximizing  $x \mapsto a^\top x + g(b^\top x)$  over  $\mathcal{X}$  is  $\mathcal{NP}$ -hard.



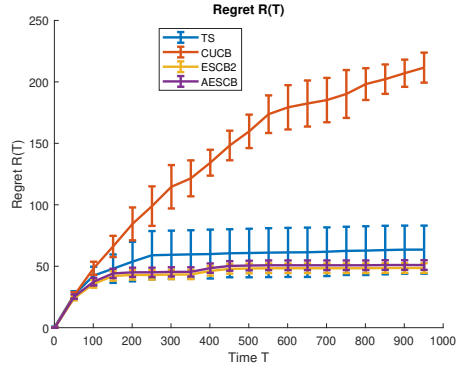
(a)  $m$ -sets,  $d = 10$



(b) spanning trees,  $|V| = 5$



(c) paths,  $|V| = 10$



(d) matchings,  $|V| = 10$

Figure 1: Expected regret of algorithms

Table 3: Computing Time

| PROBLEM        | ALGO. | TIME             | [s]              |                   |
|----------------|-------|------------------|------------------|-------------------|
| <i>m</i> -SETS | ESCB  | 0.01 ± 0.00      | 0.02 ± 0.00      | 1.24 ± 0.03       |
|                | AESCB | 0.00 ± 0.00      | 0.01 ± 0.00      | 0.10 ± 0.10       |
|                | CUCB  | 0.00 ± 0.00      | 0.00 ± 0.00      | 0.00 ± 0.00       |
|                | TS    | 0.00 ± 0.00      | 0.00 ± 0.00      | 0.00 ± 0.00       |
| PATHS          |       | ( <i>d</i> = 10) | ( <i>d</i> = 20) | ( <i>d</i> = 50)  |
|                | ESCB  | 0.00 ± 0.00      | 0.02 ± 0.00      | 0.11 ± 0.04       |
|                | AESCB | 0.00 ± 0.00      | 0.00 ± 0.00      | 0.05 ± 0.00       |
|                | CUCB  | 0.00 ± 0.00      | 0.00 ± 0.00      | 0.01 ± 0.00       |
|                | TS    | 0.00 ± 0.00      | 0.00 ± 0.00      | 0.01 ± 0.00       |
| TREES          |       | ( <i>d</i> = 10) | ( <i>d</i> = 45) | ( <i>d</i> = 190) |
|                | ESCB  | 0.02 ± 0.00      | 0.06 ± 0.05      | 0.20 ± 0.03       |
|                | AESCB | 0.00 ± 0.00      | 0.02 ± 0.00      | 0.04 ± 0.01       |
|                | CUCB  | 0.00 ± 0.00      | 0.00 ± 0.00      | 0.01 ± 0.00       |
|                | TS    | 0.00 ± 0.00      | 0.00 ± 0.00      | 0.01 ± 0.00       |
| MATCHINGS      |       | ( <i>d</i> = 10) | ( <i>d</i> = 45) | ( <i>d</i> = 190) |
|                | ESCB  | 0.01 ± 0.00      | 0.26 ± 0.05      |                   |
|                | AESCB | 0.00 ± 0.00      | 0.18 ± 0.01      |                   |
|                | CUCB  | 0.00 ± 0.00      | 0.00 ± 0.00      |                   |
|                | TS    | 0.00 ± 0.00      | 0.00 ± 0.00      |                   |
|                |       | ( <i>d</i> = 4)  | ( <i>d</i> = 25) |                   |

**Proposition 3** (*Proposition 4 of [3]*) Consider  $G = (V, E)$ , a complete bipartite graph. Consider  $\mathcal{X}$  the set of matchings in  $G$  and  $g$  a strictly concave function. Then maximizing  $x \mapsto a^\top x + g(b^\top x)$  over  $\mathcal{X}$  is  $\mathcal{NP}$ -hard.

In summary, when  $\mathcal{X}$  is either the set of  $m$ -sets, the set of paths in a directed acyclic graph, or the set of matchings in a complete bipartite graph, then  $P_2$  is  $\mathcal{NP}$ -hard.

## B Regret of Thompson Sampling

The authors of [26] study a slightly more general problem than ours, and propose a regret bound for Thompson Sampling. Let us rephrase their bound with our notations. Their problem is a combinatorial semi bandit problem with a non linear reward function. Namely they consider decisions  $x \in \mathcal{X} \subset \{0, 1\}^d$  in a combinatorial set, and the expected reward of decision  $x \in \mathcal{X}$  is given by a possibly non-linear function  $r(\theta, x)$ , where  $\theta$  is a vector unknown to the learner. The expected reward function  $r$  must satisfy the Lipschitz condition:

$$|r(\theta, x) - r(\theta', x)| \leq B|\theta'^\top x - \theta^\top x|, \forall \theta, \theta'$$

with  $B$  the Lipschitz constant. In our setting the reward function is  $r(\theta, x) = \theta^\top x$  so that  $B = 1$ .

Their main result is [26][Theorem 1], which is the regret upper bound for Thompson Sampling:

$$R(T) \leq C_5(\theta, \mathcal{X}, \varepsilon) + 8B^2(\ln T) \sum_{i=1}^d \max_{x \in \mathcal{X}: x_i=1, \Delta_x > 0} \left( \frac{\mathbf{1}^\top x}{\Delta_x - 2B(\mathbf{1}^\top x^* + 2)\varepsilon} \right) \quad \forall \varepsilon > 0$$

where  $C_5(\theta, \mathcal{X}, \varepsilon)$  is a positive number which does not depend on  $T$ . In our setting the reward function is  $r(\theta, x) = \theta^\top x$  so that  $B = 1$  and in the worse case, we will have for all  $i$ :

$$\max_{x \in \mathcal{X}: x_i=1, \Delta_x > 0} \frac{\mathbf{1}^\top x}{\Delta_x} = \frac{m}{\Delta_{\min}},$$

Hence the upper bound for the regret of Thompson Sampling provided by [26][Theorem 1] scales as:

$$R(T) = \mathcal{O} \left\{ \frac{dm \ln T}{\Delta_{\min}} \right\}, T \rightarrow \infty$$

Table 4: Regret with Parameter Tuning

| Problem                   | Algo. | Regret            |                                     |                                    |                                     |                                     |
|---------------------------|-------|-------------------|-------------------------------------|------------------------------------|-------------------------------------|-------------------------------------|
|                           |       | $(\alpha = 0.1)$  | $(\alpha = 0.2)$                    | $(\alpha = 0.3)$                   | $(\alpha = 0.4)$                    | $(\alpha = 0.5)$                    |
| $m$ -sets<br>( $d = 50$ ) | ESCB  | $31.6 \pm 24.8$   | $27.9 \pm 17.9$                     | <b><math>24.9 \pm 14.3</math></b>  | $25.1 \pm 13.5$                     | <b><math>24.9 \pm 13.0</math></b>   |
|                           | AESCB | $36.1 \pm 25.7$   | $32.5 \pm 15.6$                     | $29.6 \pm 15.1$                    | $28.7 \pm 15.4$                     | <b><math>27.4 \pm 14.1</math></b>   |
|                           | CUCB  | $164.8 \pm 129.9$ | $73.6 \pm 86.8$                     | $27.5 \pm 10.3$                    | <b><math>16.6 \pm 5.1</math></b>    | $24.8 \pm 13.9$                     |
| Paths<br>( $d = 190$ )    | ESCB  | $276.4 \pm 3.6$   | $275.0 \pm 4.5$                     | $275.0 \pm 4.5$                    | <b><math>275.0 \pm 4.4</math></b>   | <b><math>275.0 \pm 4.4</math></b>   |
|                           | AESCB | $284.0 \pm 3.8$   | $283.2 \pm 6.1$                     | $283.2 \pm 7.1$                    | <b><math>282.3 \pm 4.6</math></b>   | <b><math>282.3 \pm 4.6</math></b>   |
|                           | CUCB  | $95.9 \pm 43.0$   | $74.4 \pm 13.4$                     | <b><math>71.7 \pm 2.1</math></b>   | $84.5 \pm 8.9$                      | $102.2 \pm 5.7$                     |
| Trees<br>( $d = 190$ )    | ESCB  | $138.4 \pm 54.3$  | $137.1 \pm 51.6$                    | <b><math>128.1 \pm 37.9</math></b> | $148.1 \pm 33.3$                    | $130.1 \pm 33.8$                    |
|                           | AESCB | $166.0 \pm 57.9$  | $166.0 \pm 52.9$                    | $155.2 \pm 38.9$                   | $179.2 \pm 35.2$                    | <b><math>136.6 \pm 34.7</math></b>  |
|                           | CUCB  | $544.1 \pm 30.6$  | $225.8 \pm 79.4$                    | <b><math>205.5 \pm 73.9</math></b> | $290.5 \pm 38.7$                    | $327.8 \pm 47.2$                    |
| Matchings<br>( $d = 25$ ) | ESCB  | $108.9 \pm 119.4$ | $108.1 \pm 114.2$                   | $108.1 \pm 114.4$                  | <b><math>325.1 \pm 107.9</math></b> | <b><math>325.1 \pm 107.9</math></b> |
|                           | AESCB | $360.2 \pm 123.3$ | $360.1 \pm 123.7$                   | $357.9 \pm 117.4$                  | $357.3 \pm 113.9$                   | <b><math>357.0 \pm 113.3</math></b> |
|                           | CUCB  | $838.2 \pm 423.4$ | <b><math>365.4 \pm 104.2</math></b> | $431.1 \pm 60.6$                   | $477.9 \pm 106.6$                   | $566.9 \pm 76.7$                    |

and this bound does not match the (smaller) regret upper bound of algorithms such as ESCB and AESCB which is

$$R(T) = \mathcal{O} \left\{ \frac{d(\ln m)^2 \ln T}{\Delta_{\min}} \right\}, T \rightarrow \infty$$

## C Casting Optimization problem $(P_2)$ as a ISOCP

The optimization problem  $(P_2)$ , whose definition is:

$$\underset{x}{\text{maximize}} \{a^\top x + \sqrt{b^\top x}\} \text{ subject to } x \in \mathcal{X} \quad (P_2)$$

can be cast as an Integer Second-Order Cone Program (MISOCP), which enables one to solve it using a standard ISOP solver. Indeed, the objective function features a geometric mean, which a special case of hyperbolic constraint [18]. Problem  $(P_2)$  can be rewritten as:

$$\underset{(x,t)}{\text{maximize}} \{a^\top x + t\} \text{ subject to } x \in \mathcal{X} \quad \text{and} \quad t^2 \leq b^\top x$$

Applying the transformation proposed in [18][Section 2.3], the hyperbolic constraint can be written as a SOCP (Second-Order Cone Program):

$$\underset{(x,t)}{\text{maximize}} \{a^\top x + t\} \text{ subject to } x \in \mathcal{X} \quad \text{and} \quad \left\| \begin{bmatrix} 2t \\ b^\top x - 1 \end{bmatrix} \right\| \leq b^\top x + 1$$

Even though the constraints defining  $\mathcal{X}$  ensure that optimizing a linear objective over  $\mathcal{X}$  yields an integer solution, this is no more the case with the new formulation. Hence, integrality constraints must be added for the relevant variables.

$$\underset{(x,t)}{\text{maximize}} \{a^\top x + t\} \text{ subject to } x \in \mathcal{X} \quad \text{and} \quad \left\| \begin{bmatrix} 2t \\ b^\top x - 1 \end{bmatrix} \right\| \leq b^\top x + 1 \quad \text{and} \quad x \in \{0, 1\}^d$$

This formulation is an ISOCP and can be readily solved to optimality by existing software such as CPLEX, Gurobi, Pajarito [11], Mosek, or Xpress.

## D Proof of Theorem 4.4

**Theorem 4.4.** The regret of AESCB with parameters  $(\varepsilon_t, \delta_t)$  admits the following upper bound for all  $T \geq 1$ :

$$R(T) \leq C_4(m) + \frac{2dm^3}{\Delta_{\min}^2} + \frac{24df(T)}{(\min_{t \leq T} \varepsilon_t)^2 \Delta_{\min}} \left\lceil \frac{\ln m}{1.61} \right\rceil^2 + 4 \sum_{t=1}^T \delta_t \mathbf{1}(\Delta_{\min} \leq 4\delta_t).$$

with  $f(t) = \ln t + 4m \ln \ln t$  and  $C_4(m)$  a positive number that solely depends on  $m$ .

By corollary, for  $\varepsilon_t = \varepsilon$  and  $\lim_{t \rightarrow \infty} \delta_t = 0$ , we have:

$$R(T) = \mathcal{O}\left(d(\ln m)^2 \frac{1}{\Delta_{\min}} \ln T\right) \quad \text{as } T \rightarrow \infty.$$

Similarly, with  $\varepsilon_t = \varepsilon$  and  $\delta_t < \frac{1}{4}\Delta_{\min}$ , we have, for all  $T \geq 1$ :

$$R(T) \leq C_4(m) + \frac{2dm^3}{\Delta_{\min}^2} + \frac{24df(T)}{\varepsilon^2 \Delta_{\min}} \left\lceil \frac{\ln m}{1.61} \right\rceil^2.$$

**Proof.** We decompose the regret based on three events:

- $G_t$ : the estimate  $\hat{\theta}(t)$  deviates abnormally from  $\theta$  so that:

$$G_t = \left\{ \theta^\top x^\star \geq \hat{\theta}(t)^\top x^\star + \sqrt{\sigma^2(t)^\top x^\star} \right\}.$$

- $H_t$ : the reward of the decision chosen at time  $t$  is poorly estimated, namely:

$$H_{i,t} = \left\{ x_i(t) = 1, \quad \left| \hat{\theta}_i(t) - \theta_i \right| \geq \frac{\Delta_{\min}}{2m} \right\} \quad \text{and} \quad H_t = \bigcup_{i=1}^d H_{i,t}.$$

- $I_t$ : the reward gap of the decision chosen at time  $t$  is small

$$I_t = \left\{ \Delta_{x(t)} \leq 4\delta_t \right\}$$

Of course, most of the time,  $\overline{G}_t$  and  $\overline{H}_t$  occur, since both  $G_t$  and  $H_t$  have small probability. Therefore,  $G_t$  and  $H_t$  cause only a constant regret as we shall see. Also,  $I_t$  causes a regret that is at most  $4\delta_t$ . For all  $x \in \mathcal{X}$  and  $t \geq 1$ , we define the exploration bonus  $E_t(x)$  of decision  $x$  at time  $t$ :

$$E_t(x) = \sqrt{\sigma^2(t)^\top x}.$$

**Generic regret bound.** Recall that the regret is defined as:

$$R(T) = \mathbb{E} \left\{ \sum_{t=1}^T \Delta_{x(t)} \right\} = \mathbb{E} \left\{ \sum_{t=1}^T \Delta_{x(t)} \mathbf{1}\{x(t) \neq x^\star\} \right\},$$

Decomposing according to the occurrence of  $G_t$  and  $H_t$ , we get:

$$\begin{aligned} R(T) &\leq \mathbb{E} \left\{ \sum_{t=1}^T \mathbf{1}\{G_t\} \Delta_{x(t)} \right\} + \mathbb{E} \left\{ \sum_{t=1}^T \mathbf{1}\{H_t\} \Delta_{x(t)} \right\} + \mathbb{E} \left\{ \sum_{t=1}^T \mathbf{1}\{I_t\} \Delta_{x(t)} \right\} \\ &\quad + \mathbb{E} \left\{ \sum_{t=1}^T \mathbf{1}\{\overline{G}_t, \overline{H}_t, \overline{I}_t, x(t) \neq x^\star\} \Delta_{x(t)} \right\}. \end{aligned}$$

Define  $\bar{\varepsilon}_T = \min_{t \leq T} \varepsilon_t$ . The last term can be rewritten in terms of the following event:

$$F_t = \left\{ \Delta_{x(t)} \leq \frac{4}{\bar{\varepsilon}_T} E_t(x(t)) \right\},$$

Let us prove that  $(\bar{G}_t \cap \bar{H}_t \cap \bar{I}_t \cap \{x(t) \neq x^*\}) \subset F_t$ . Assume that  $(\bar{G}_t \cap \bar{H}_t \cap \bar{I}_t \cap \{x(t) \neq x^*\})$  occurs. Then,

$$\begin{aligned} \theta^\top x^* &\leq \hat{\theta}(t)^\top x^* + E_t(x^*) \\ &\leq \max_{x \in \mathcal{X}} \{\hat{\theta}(t)^\top x + E_t(x)\} \\ &\leq \delta_t + \hat{\theta}(t)^\top x(t) + \frac{1}{\varepsilon_t} E_t(x(t)) \\ &\leq \delta_t + \theta^\top x(t) + \frac{\Delta_{x(t)}}{2} + \frac{1}{\varepsilon_t} E_t(x(t)) \\ &\leq \theta^\top x(t) + \frac{3\Delta_{x(t)}}{4} + \frac{1}{\varepsilon_t} E_t(x(t)) \end{aligned}$$

where we successively used the fact that  $\bar{G}_t$  occurs, the definition of AESCB, the fact that  $\hat{\theta}(t)^\top x(t) \leq \theta^\top x(t) + \frac{\Delta_{x(t)}}{2}$  since  $\bar{H}_t$  occurs, and the fact that  $\delta_t \leq \frac{\Delta_{x(t)}}{4}$  since  $\bar{I}_t$  occurs. Therefore,

$$\frac{\Delta_{x(t)}}{4} \leq \frac{1}{\varepsilon_t} E_t(x(t)) \leq \frac{1}{\bar{\varepsilon}_T} E_t(x(t))$$

so that  $F_t$  indeed occurs.

As a consequence, the regret is upper bounded by the sum of four terms:

$$R(T) \leq \mathbb{E} \left\{ \sum_{t=1}^T \mathbf{1}\{G_t\} \Delta_{x(t)} \right\} + \mathbb{E} \left\{ \sum_{t=1}^T \mathbf{1}\{H_t\} \Delta_{x(t)} \right\} + \mathbb{E} \left\{ \sum_{t=1}^T \mathbf{1}\{I_t\} \Delta_{x(t)} \right\} + \mathbb{E} \left\{ \sum_{t=1}^T \mathbf{1}\{F_t\} \Delta_{x(t)} \right\}$$

**First term: poor reward estimation.** Recall that for any  $x$  we have  $\Delta_x \leq \theta^\top x^* \leq m$ , since  $\theta \in [0, 1]^d$  and  $\max_{x \in \mathcal{X}} \mathbf{1}^\top x = m$ . Therefore, by applying [12][Theorem 3]:

$$\mathbb{E} \left\{ \sum_{t=1}^T \mathbf{1}\{G_t\} \Delta_{x(t)} \right\} \leq m \mathbb{E} \left\{ \sum_{t=1}^T \mathbf{1}\{G_t\} \right\} = m \sum_{t=1}^{\infty} \mathbb{P}(G_t) \leq C_4(m).$$

where  $C_4(m)$  is a positive number which only depends on  $m$ , as stated by [12][Theorem 3].

**Second term: poor choice of item.** We turn to the second term, using a union bound

$$\mathbb{P}(H_t) = \mathbb{P} \left( \bigcup_{i=1}^d H_{t,i} \right) \leq \sum_{i=1}^d \mathbb{P}(H_{t,i}).$$

Using once again the fact that  $\Delta_{x(t)} \leq m$ , the regret due to  $H_t = \bigcup_{i=1}^d H_{t,i}$  is bounded by using a union bound:

$$\mathbb{E} \left\{ \sum_{t=1}^T \mathbf{1}\{H_t\} \Delta_{x(t)} \right\} \leq m \mathbb{E} \left\{ \sum_{t=1}^T \mathbf{1}\{H_t\} \right\} = m \sum_{t=1}^T \mathbb{P}(H_t) \leq m \sum_{t=1}^T \sum_{i=1}^d \mathbb{P}(H_{t,i}).$$

By definition of  $H_{t,i}$ , we get

$$\mathbb{E} \left\{ \sum_{t=1}^T \mathbf{1}\{H_t\} \Delta_{x(t)} \right\} \leq m \sum_{t=1}^T \sum_{i=1}^d \mathbb{P} \left( x_i(t) = 1, \quad |\hat{\theta}_i(t) - \theta_i| \geq \frac{\Delta_{x(t)}}{2m} \right)$$

Using Hoeffding's inequality, this probability can be bounded by:

$$\mathbb{E} \left\{ \sum_{t=1}^T \mathbf{1}\{H_t, x(t) \neq x^*\} \Delta_{x(t)} \right\} \leq m \sum_{t=1}^T \sum_{i=1}^d e^{-t \left( \frac{\Delta_{\min}}{m} \right)^2} \leq \frac{md}{1 - e^{-\left( \frac{\Delta_{\min}}{m} \right)^2}} \leq \frac{m^3 d (1 + \frac{\Delta_{\min}^2}{m^2})}{\Delta_{\min}^2} \leq \frac{2m^3 d}{\Delta_{\min}^2}$$

where we recognize a geometric series and use the elementary inequality  $e^z \geq 1 + z$  for all  $z$ , which gives  $e^{-z} \leq (1 + z)^{-1}$  and  $(1 - e^{-z})^{-1} \leq (1 + z)z^{-1}$ . We also used the fact that  $\Delta_{\min} \leq m$ .

**Third term: small reward gap.** By definition, the third term is:

$$\mathbb{E} \left\{ \sum_{t=1}^T \mathbf{1}\{I_t\} \Delta_{x(t)} \right\} = \mathbb{E} \left\{ \sum_{t=1}^T \mathbf{1}\{\Delta_{x(t)} \leq 4\delta_t\} \Delta_{x(t)} \right\} \leq 4 \sum_{t=1}^T \delta_t \mathbf{1}\{\Delta_{\min} \leq 4\delta_t\}.$$

**Fourth term: dominant term.** We now consider the event  $\Delta_{x(t)} \leq \frac{4}{\varepsilon_T} E_t(x(t))$ . Squaring, we get:

$$\Delta_{x(t)}^2 \leq \frac{16}{\varepsilon_T^2} E_t(x(t))^2 = \frac{16}{\varepsilon_T^2} \sigma^2(t)^\top x(t) = \frac{8f(t)}{\varepsilon_T^2} \sum_{i=1}^d \frac{x_i}{n_i(t)}.$$

where we used the definition of  $E_t$  and that of  $\sigma^2(t)$ . If this event happens, it means that there exists a subset of indices  $i = 1, \dots, d$  such that the number of samples  $n_i(t)$  is small. We further decompose this event as follows.

Consider  $(\alpha_j)_{j \in \mathbb{N}}$  and  $(\beta_j)_{j \in \mathbb{N}}$ , two positive, non-increasing sequences verifying the following properties:

- $\lim_{j \rightarrow \infty} \alpha_j = \lim_{j \rightarrow \infty} \beta_j = 0$
- $\lim_{j \rightarrow +\infty} \frac{\beta_j}{\sqrt{\alpha_j}} = 0$
- $\beta_0 = 1$

We can define  $j_0$  as the first integer  $j$  such that  $\beta_j \leq \frac{1}{m}$  and define  $l$  as the sum

$$l = \frac{\beta_{j_0}}{\alpha_{j_0}} + \sum_{j=1}^{j_0} \frac{\beta_{j-1} - \beta_j}{\alpha_{j-1}},$$

Sequences  $(\alpha_j)_{j \in \mathbb{N}}$  and  $(\beta_j)_{j \in \mathbb{N}}$  are fixed, and their exact value will be specified later.

For all  $j \in \mathbb{N}$ , define the following sets:

$$S_t^j = \left\{ i \in \{1, \dots, d\} : x_i(t) = 1, n_i(t) \leq \alpha_j \frac{2f(t)g(m)}{\Delta_{x(t)}^2} \right\} \text{ if } j \geq 1 \text{ and } S_0^j = \{i \in \{1, \dots, d\} : x_i(t) = 1\} \text{ otherwise}$$

where  $g(m)$  is defined as  $g(m) = \frac{4ml}{\varepsilon_T}$  with  $l$  a constant to be defined later.

Since  $j \mapsto \alpha_j$  is decreasing and  $\lim_{j \rightarrow \infty} \alpha_j = 0$ , the sequence  $S_t^j$  is decreasing for set inclusion; moreover, there is an index  $j_0$  such that  $S_n^{j_0} = \emptyset$ :

$$\emptyset = S_t^{j_0} \subset S_t^{j_0-1} \subset \dots \subset S_t^1 \subset S_t^0$$

Define the event  $A_t^j$  as:

$$A_t^j = \left\{ |S_t^j| \geq m \beta_j \quad \text{and} \quad \forall k < j, \quad |S_t^k| < m \beta_k \right\}.$$

By assumption, we have:

$$|S_t^0| = m \beta_0 = m.$$

Finally, also define the events  $A_t$  as the following unions:

$$A_t = \bigcup_{j=1}^{+\infty} A_t^j$$

We have that  $A_t$  is a finite union of events  $A_t^j$ :

$$A_t = \bigcup_{j=1}^{j_0} A_t^j.$$

Indeed, for all  $j > j_0$ , due to  $\beta_{j_0} \leq 1/m$  and the fact that  $\beta_j$  is a decreasing sequence,

$$m \beta_j \leq m \frac{1}{m} = 1.$$

Thus, by definition of the event  $A_t^j$ , we have that:

$$A_t^j = \left\{ |S_t^j| \geq 1 \quad \text{and} \quad \forall k < j_0, \quad |S_t^j| < m \beta_j \quad \text{and} \quad \forall k \in [j_0, j-1], \quad |S_t^j| = 0 \right\}.$$

However, the same set  $S_t^j$  cannot be both empty and have at least one element. In other words, the event  $A_t^j$  cannot happen for  $j > j_0$ .

Under event  $\bar{A}_t$ , the sum  $\sum_{i=1}^d \frac{x_i(t)}{n_i(t)}$  can be bounded. The event  $\bar{A}_t$  is, by De Morgan's law (recall that  $j_0$  is finite):

$$\begin{aligned} \bar{A}_t &= \bigcap_{j=1}^{j_0} \bar{A}_t^j \\ &= \bigcap_{j=1}^{j_0} \left\{ |S_t^j| < m \beta_j \quad \text{or} \quad \exists k < j, \quad |S_t^j| \geq m \beta_j \right\}, \quad \text{by definition of } A_t^j \\ &= \bigcap_{j=1}^{j_0} \left[ \left\{ |S_t^j| < m \beta_j \right\} \cup \left\{ \bigcup_{k=1}^{j-1} |S_t^j| \geq m \beta_j \right\} \right] \\ &= \bigcap_{j=1}^{j_0} \left\{ |S_t^j| < m \beta_j \right\}, \quad \text{as the latter events are not possible if the first holds} \\ &= \bigcap_{j=1}^{j_0-1} \left\{ |S_t^j| < m \beta_j \right\} \cap \left\{ |S_t^{j_0}| < m \beta_{j_0} \right\}. \end{aligned}$$

Since  $\beta_{j_0} \leq 1/m$ , the last event can be written as  $|S_t^{j_0}| < \frac{m}{m} = 1$ . A set whose cardinality is strictly less than one must be empty, thus:

$$\bar{A}_t = \bigcap_{j=1}^{j_0-1} \left\{ |S_t^j| < m \beta_j \right\} \cap \left\{ |S_t^{j_0}| = 0 \right\}.$$

If event  $\bar{A}_t$  happens, then:

$$\begin{aligned} \bar{S}_t^j &= \left\{ i = 1, \dots, d : x_i(t) = 1, i \notin S_t^j \right\} = \left\{ i = 1, \dots, d : x_i(t) = 1, n_i(t) > \alpha_j \frac{2f(t)}{\Delta_{x(t)}^2} g(m) \right\}, \\ \bar{S}_t^{j_0} &= \{i = 1, \dots, d : x_i(t) = 1\}. \end{aligned}$$

Indeed, due to the fact that  $S_t^j$  is a decreasing sequence for set inclusion, the complement  $\bar{S}_t^j$  must be an increasing sequence for set inclusion. This implies that:

$$\{i = 1, \dots, d : x_i(t) = 1\} = \bigcup_{j=1}^{j_0} (\bar{S}_t^j \setminus \bar{S}_t^{j-1}).$$

Thus,

$$\sum_{i=1}^d \frac{x_i(t)}{n_i(t)} = \sum_{j=1}^{j_0} \sum_{i \in \bar{S}_t^j \setminus \bar{S}_t^{j-1}} \frac{x_i(t)}{n_i(t)}.$$

Using the definition of  $\bar{S}_t^j$ , one might write that, if  $\bar{S}_t^j$  holds, then:

$$\sum_{i \in \bar{S}_t^j \setminus \bar{S}_t^{j-1}} \frac{x_i(t)}{n_i(t)} < \frac{\Delta_{x(t)}^2}{2f(t)g(m)\alpha_j} \sum_{i \in \bar{S}_t^j \setminus \bar{S}_t^{j-1}} x_i(t) = \frac{\Delta_{x(t)}^2}{2f(t)g(m)} \frac{|\bar{S}_t^j \setminus \bar{S}_t^{j-1}|}{\alpha_j}.$$

This implies that the previous sum is bounded by:

$$\sum_{i=1}^d \frac{x_i(t)}{n_i(t)} = \sum_{j=1}^{j_0} \sum_{i \in \bar{S}_t^j \setminus \bar{S}_t^{j-1}} \frac{x_i(t)}{n_i(t)} \leq \frac{\Delta_{x(t)}^2}{2f(t)g(m)} \sum_{j=1}^{j_0} \frac{|\bar{S}_t^j \setminus \bar{S}_t^{j-1}|}{\alpha_j}$$

The inner sum can be decomposed as follows, by definition of  $S_t^j$  and  $\bar{S}_t^j$ :

$$\begin{aligned} \sum_{j=1}^{j_0} \frac{|\bar{S}_t^j \setminus \bar{S}_t^{j-1}|}{\alpha_j} &= \sum_{j=1}^{j_0} \frac{|S_t^j \setminus S_t^{j-1}|}{\alpha_j}, \quad \text{dropping the complements} \\ &= \sum_{j=1}^{j_0} \frac{|S_t^j| - |S_t^{j-1}|}{\alpha_j} \\ &= \frac{|S_t^{j_0}|}{\alpha_0} + \sum_{j=1}^{j_0} \left[ |S_t^j| \left( \frac{1}{\alpha_{j-1}} - \frac{1}{\alpha_j} \right) \right], \quad \text{by factoring the last term } j_0 \\ &< \frac{m\beta_{j_0}}{\alpha_0} + \sum_{j=1}^{j_0} \left[ m\beta_j \left( \frac{1}{\alpha_{j-1}} - \frac{1}{\alpha_j} \right) \right], \quad \text{as } \bar{A}_t \text{ holds.} \end{aligned}$$

Finally, replacing  $g$  and  $l$  by their definition

$$\sum_{i=1}^d \frac{x_i(t)}{n_i(t)} < \frac{m\Delta_{x(t)}^2}{2f(t)g(m)} \left( \frac{\beta_{j_0}}{\alpha_{j_0}} + \sum_{j=1}^{j_0} \frac{\beta_{j-1} - \beta_j}{\alpha_{j-1}} \right) = \frac{\Delta_{x(t)}^2 \bar{\varepsilon}_T^2}{8f(t)}$$

We prove that event  $\Delta_{x(t)} \leq \frac{4}{\bar{\varepsilon}_T} E_t(x(t))$  implies  $A_t$ . Indeed, if  $\Delta_{x(t)} \leq \frac{4}{\bar{\varepsilon}_T} E_t(x(t))$  and  $\bar{A}_t$ , then

$$\Delta_{x(t)}^2 \leq \frac{16}{\bar{\varepsilon}_T^2} E_t(x(t))^2 = \frac{16}{\bar{\varepsilon}_T^2} \sigma^2(t)^\top x(t) = \frac{8f(t)}{\bar{\varepsilon}_T^2} \sum_{i=1}^d \frac{x_i}{n_i(t)} < \Delta_{x(t)}^2$$

which is a contradiction. Therefore,  $\Delta_{x(t)} \leq \frac{4}{\bar{\varepsilon}_T} E_t(x(t))$  implies  $A_t$ .

We now bound the regret due to the event  $A_t$ . We further decompose  $A_t^j$  to include the fact that a specific item  $i$  is included among the (at least)  $m\beta_j$  items that have not yet been selected enough:

$$A_t^{j,i} = A_t^j \cap \left\{ x_i(t) = 1, n_i(t) \leq \frac{\alpha_j 2f(T)g(m)}{\Delta_{x(t)}^2} \right\}.$$

Of course, the union over all  $i$  yields back  $A_t^j$ :

$$\bigcup_{i=1}^d A_t^{j,i} = A_t^j.$$

Since  $A_t^j$  implies that at least  $m\beta_j$  items have not yet been selected enough,

$$\mathbf{1}\{A_t^j\} \leq \frac{1}{m\beta_j} \sum_{i=1}^d \mathbf{1}\{A_t^{j,i}\}.$$

The contribution to the regret is bounded by the items that are not selected frequently enough to ensure a good reward estimate:

$$\sum_{t=1}^T \Delta_{x(t)} \mathbf{1}\{F_t\} \leq \sum_{t=1}^T \Delta_{x(t)} \mathbf{1}\{A_t\} \leq \sum_{t=1}^T \sum_{j=1}^{+\infty} \Delta_{x(t)} \mathbf{1}\{A_t^j\} \leq \sum_{t=1}^T \sum_{j=1}^{+\infty} \sum_{i=1}^d \frac{\Delta_{x(t)}}{m\beta_j} \mathbf{1}\{A_t^{j,i}\}.$$

For any  $i$ , define the possible values of the gaps  $\Delta_x$  where  $x_i = 1$ , namely:

$$\{\Delta_x : x \in \mathcal{X}, x_i = 1\} = \{\Delta_{i,1}, \dots, \Delta_{i,K_i}\}$$

where  $K_i$  is the number of possible values for the gap and we assume that the gaps are sorted in decreasing order:

$$\Delta_{i,1} > \dots > \Delta_{i,K_i}$$

with the convention that  $\Delta_{i,0} = \infty$ . We can then decompose the previous sum according to the value of the gap:

$$\begin{aligned} \sum_{t=1}^T \mathbf{1}\{F_t\} \Delta_{x(t)} &\leq \sum_{t=1}^T \sum_{j=1}^{+\infty} \sum_{i=1}^d \frac{\Delta_{x(t)}}{m\beta_j} \mathbf{1}\{A_t^{j,i}\} \\ &\leq \sum_{t=1}^T \sum_{j=1}^{+\infty} \sum_{i=1}^d \sum_{k=1}^{K_i} \frac{\Delta_{i,k}}{m\beta_j} \mathbf{1}\{A_t^{j,i}, \Delta_{x(t)} = \Delta_{i,k}\} \\ &\leq \sum_{t=1}^T \sum_{j=1}^{+\infty} \sum_{i=1}^d \sum_{k=1}^{K_i} \frac{\Delta_{i,k}}{m\beta_j} \mathbf{1}\left\{A_t^j, x_i(t) = 1, n_i(t) \leq \frac{\alpha_j f(T) g(m)}{2\Delta_{i,k}^2}, \Delta_{x(t)} = \Delta_{i,k}\right\}, \end{aligned}$$

by definition of  $A_t^{j,i}$ . To simplify notation, let

$$\tau_j = \frac{1}{2} \alpha_j f(T) g(m).$$

Thus, the previous bound can be written as:

$$\sum_{t=1}^T \mathbf{1}\{F_t\} \Delta_{x(t)} \leq \sum_{t=1}^T \sum_{j=1}^{+\infty} \sum_{i=1}^d \sum_{k=1}^{K_i} \frac{\Delta_{i,k}}{m\beta_j} \mathbf{1}\left\{i : x_i(t) = 1, \quad n_i(t) \leq \frac{\tau_j}{\Delta_{i,k}^2}, \quad \Delta_{x(t)} = \Delta_{i,k}\right\}.$$

To simplify the developments, focus on the two sums, the one on the rounds  $t$  and the one on the gaps  $k$ :

$$\sum_{t=1}^T \sum_{k=1}^{K_i} \frac{\Delta_{i,k}}{m\beta_j} \mathbf{1}\left\{x_i(t) = 1, \quad n_i(t) \leq \frac{\tau_j}{\Delta_{i,k}^2}, \quad \Delta_{x(t)} = \Delta_{i,k}\right\}$$

As the values of  $\Delta_{i,k}$  are ordered, we can decompose as follows:

$$\begin{aligned} & \sum_{t=1}^T \sum_{k=1}^{K_i} \frac{\Delta_{i,k}}{m \beta_j} \mathbf{1} \left\{ x_i(t) = 1, \quad n_i(t) \leq \frac{\tau_j}{\Delta_{i,k}^2}, \quad \Delta_{x(t)} = \Delta_{i,k} \right\} \\ &= \sum_{t=1}^T \sum_{k=1}^{K_i} \sum_{p=1}^n \frac{\Delta_{i,k}}{m \beta_j} \mathbf{1} \left\{ x_i(t) = 1, \quad n_i(t) \in \left( \frac{\tau_j}{\Delta_{i,p-1}^2}, \frac{\tau_j}{\Delta_{i,p}^2} \right], \quad \Delta_{x(t)} = \Delta_{i,k} \right\} \end{aligned}$$

The factor  $\Delta_{i,k}$  can be moved within the summation over  $j$  and be rewritten as  $\Delta_{i,k}$ , as it will be counted only once, when the step function is nonzero (when  $j = k$ ):

$$\begin{aligned} & \sum_{t=1}^T \sum_{k=1}^{K_i} \frac{\Delta_{i,k}}{m \beta_j} \mathbf{1} \left\{ x_i(t) = 1, \quad n_i(t) \leq \frac{\tau_j}{\Delta_{i,k}^2}, \quad \Delta_{x(t)} = \Delta_{i,k} \right\} \\ & \leq \sum_{t=1}^T \sum_{k=1}^{K_i} \sum_{p=1}^n \frac{\Delta_{i,p}}{m \beta_j} \mathbf{1} \left\{ x_i(t) = 1, \quad n_i(t) \in \left( \frac{\tau_j}{\Delta_{i,p-1}^2}, \frac{\tau_j}{\Delta_{i,p}^2} \right], \quad \Delta_{x(t)} = \Delta_{i,k} \right\} \end{aligned}$$

If the sum over  $p$  goes to  $K_i$ , many new terms can be added, though:

$$\begin{aligned} & \sum_{t=1}^T \sum_{k=1}^{K_i} \frac{\Delta_{i,k}}{m \beta_j} \mathbf{1} \left\{ x_i(t) = 1, \quad n_i(t) \leq \frac{\tau_j}{\Delta_{i,k}^2}, \quad \Delta_{x(t)} = \Delta_{i,k} \right\} \\ & \leq \sum_{t=1}^T \sum_{k=1}^{K_i} \sum_{p=1}^{K_i} \frac{\Delta_{i,p}}{m \beta_j} \mathbf{1} \left\{ x_i(t) = 1, \quad n_i(t) \in \left( \frac{\tau_j}{\Delta_{i,p-1}^2}, \frac{\tau_j}{\Delta_{i,p}^2} \right], \quad \Delta_{x(t)} = \Delta_{i,k} \right\} \end{aligned}$$

Again, if the solution  $x(t)$  is not taken to be exactly  $k$ , but any suboptimal solution, many new terms now count in the summation. With this change, the sum over  $k$  becomes irrelevant, as all gaps that may contribute to the regret are still counted in.

$$\begin{aligned} & \sum_{t=1}^T \sum_{k=1}^{K_i} \frac{\Delta_{i,k}}{m \beta_j} \mathbf{1} \left\{ x_i(t) = 1, \quad n_i(t) \leq \frac{\tau_j}{\Delta_{i,k}^2}, \quad \Delta_{x(t)} = \Delta_{i,k} \right\} \\ & \leq \sum_{t=1}^T \sum_{p=1}^{K_i} \frac{\Delta_{i,p}}{m \beta_j} \mathbf{1} \left\{ x_i(t) = 1, \quad n_i(t) \in \left( \frac{\tau_j}{\Delta_{i,p-1}^2}, \frac{\tau_j}{\Delta_{i,p}^2} \right], \quad \Delta_{x(t)} > 0 \right\} \\ & \leq \frac{\tau_j}{m \beta_j} \left( \frac{1}{\Delta_{i,1}} + \sum_{p=2}^{K_i} \Delta_{i,p} \left( \frac{1}{\Delta_{i,p}^2} - \frac{1}{\Delta_{i,p-1}^2} \right) \right) \leq \frac{2\tau_j}{m \beta_j \Delta_{\min}} \end{aligned}$$

where we used the following algebra, since  $\Delta_{i,1} > \dots > \Delta_{i,K_i}$ :

$$\begin{aligned} \frac{1}{\Delta_{i,1}} + \sum_{p=2}^{K_i} \Delta_{i,p} \left( \frac{1}{\Delta_{i,p}^2} - \frac{1}{\Delta_{i,p-1}^2} \right) &= \frac{1}{\Delta_{i,K_i}} + \sum_{p=1}^{K_i-1} \frac{\Delta_{i,p} - \Delta_{i,p+1}}{\Delta_{i,p}^2} \leq \frac{1}{\Delta_{i,K_i}} + \sum_{p=1}^{K_i-1} \frac{\Delta_{i,p} - \Delta_{i,p+1}}{\Delta_{i,p+1} \Delta_{i,p}} \\ &= \frac{1}{\Delta_{i,K_i}} + \sum_{p=1}^{K_i-1} \frac{1}{\Delta_{i,p+1}} - \frac{1}{\Delta_{i,p}} = \frac{2}{\Delta_{i,K_i}} - \frac{1}{\Delta_{i,1}} \leq \frac{2}{\Delta_{\min}} \end{aligned}$$

Injecting this result into the regret term bound,

$$\begin{aligned}
\sum_{t=1}^T \mathbf{1}\{F_t\} \Delta_{x(t)} &\leq \sum_{t=1}^T \sum_{j=1}^{+\infty} \sum_{i=1}^d \sum_{k=1}^{K_i} \frac{\Delta_{i,k}}{m \beta_j} \mathbf{1}\left\{x_i(t) = 1, \quad n_i(t) \leq \frac{\tau_j}{\Delta_{i,k}^2}, \quad \Delta_{x(t)} = \Delta_{i,k}\right\} \\
&\leq \sum_{j=1}^{+\infty} \sum_{i=1}^d \frac{2\tau_j}{m \beta_j \Delta_{\min}} \\
&= \frac{f(T) dg(m)}{m \Delta_{\min}} \left[ \sum_{j=1}^{j_0} \frac{\alpha_j}{\beta_j} \right], \quad \text{by definition of } \tau_j \\
&= \frac{4ld f(T)}{\bar{\varepsilon}_T^2 \Delta_{\min}} \left[ \sum_{j=1}^{j_0} \frac{\alpha_j}{\beta_j} \right], \quad \text{by definition of } g.
\end{aligned}$$

Now, set  $\alpha_i = \beta_i = \beta^i$ ,  $\beta \in (0, 1)$ , which satisfies the previous assumptions. Since  $j_0$  is the first integer  $j$  such that  $\beta_j \leq m^{-1}$ , we have  $j_0 = \left\lceil \frac{\ln m}{\ln \beta^{-1}} \right\rceil$ . Also,

$$\begin{aligned}
l \sum_{j=1}^{j_0} \frac{\alpha_j}{\beta_j} &= l j_0, \quad \text{as } \alpha_i = \beta_i \\
&= j_0 \left( \frac{\beta_{j_0}}{\alpha_{j_0}} + \sum_{j=1}^{j_0} \frac{\beta_{j-1} - \beta_j}{\alpha_{j-1}} \right), \quad \text{by definition of } l \\
&= j_0 \left( 1 + \sum_{j=1}^{j_0} \frac{\beta^{j-1} - \beta^j}{\beta^j} \right) \\
&= j_0 \left( 1 + \sum_{j=1}^{j_0} \frac{1 - \beta}{\beta} \right) \\
&= j_0 \left( 1 + \frac{j_0}{\beta} - j_0 \right) \\
&\leq j_0 \left( 1 + \frac{j_0}{\beta} \right).
\end{aligned}$$

Taking  $\beta = 1/5$ ,

$$j_0 = \left\lceil \frac{\ln m}{\ln \beta^{-1}} \right\rceil \leq \left\lceil \frac{\ln m}{1.61} \right\rceil.$$

Injecting these into the regret term, we get:

$$\sum_{t=1}^T \Delta_{x(t)} \mathbf{1}\{F_t\} \leq \frac{4ld f(T)}{\bar{\varepsilon}_T^2 \Delta_{\min}} \left[ \sum_{j=1}^{j_0} \frac{\alpha_j}{\beta_j} \right] \leq \frac{4d f(T)}{\bar{\varepsilon}_T^2 \Delta_{\min}} \left( \left\lceil \frac{\ln m}{1.61} \right\rceil + 5 \left\lceil \frac{\ln m}{1.61} \right\rceil^2 \right) \leq \frac{24d f(T)}{\bar{\varepsilon}_T^2 \Delta_{\min}} \left\lceil \frac{\ln m}{1.61} \right\rceil^2.$$

**Complete regret bound.** Gathering the results about the three terms of the regret decomposition, the regret can be bounded by:

$$R(T) \leq C_4(m) + \frac{2dm^3}{\Delta_{\min}^2} + \frac{24d f(T)}{\bar{\varepsilon}_T^2 \Delta_{\min}} \left\lceil \frac{\ln m}{1.61} \right\rceil^2 + 4 \sum_{t=1}^T \delta_t \mathbf{1}\{\Delta_{\min} \leq 4\delta_t\}.$$

## E Proof of Theorem 5.1

The first step is to upper bound the value of problem  $(P_2)$  with inputs  $a(t)$  and  $b(t)$  as a function of  $x(t)$ .

Define the set:

$$S(t) = \{0, \dots, m\xi(t)\}.$$

By definition,  $a(t) \in \{0, \dots, \xi(t)\}$ , and  $m = \max_{x \in \mathcal{X}} \mathbf{1}^\top x$ , so:

$$a(t)^\top x \in S(t), \forall x \in \mathcal{X}.$$

Therefore:

$$\begin{aligned} & \max_{x \in \mathcal{X}} \left\{ a(t)^\top x + \sqrt{b(t)^\top x} \right\} \\ &= \max_{s \in S} \max_{x \in \mathcal{X}, a(t)^\top x = s} \left\{ s + \sqrt{b(t)^\top x} \right\} \\ &\leq \max_{s \in S} \max_{x \in \mathcal{X}, a(t)^\top x \geq s} \left\{ s + \sqrt{b(t)^\top x} \right\} \\ &\leq \max_{s \in S} \left\{ s + \frac{1}{\varepsilon_t} \sqrt{b(t)^\top \bar{x}^s(t)} \right\} \\ &= s^*(t) + \frac{1}{\varepsilon_t} \sqrt{b(t)^\top \bar{x}^{s^*}(t)} \\ &\leq a(t)^\top \bar{x}^{s^*}(t) + \frac{1}{\varepsilon_t} \sqrt{b(t)^\top \bar{x}^{s^*}(t)} \\ &= a(t)^\top x(t) + \frac{1}{\varepsilon_t} \sqrt{b(t)^\top x(t)} \end{aligned}$$

where we used the fact that, by definition,  $\bar{x}^s(t)$  is an  $\varepsilon_t$ -approximate solution to  $(P_3)$ , that

$$s^*(t) \in \arg \max_{s \in S(t)} \left\{ s + \frac{1}{\varepsilon_t} \sqrt{b(t)^\top \bar{x}^s(t)} \right\}$$

and that  $a(t)^\top \bar{x}^s(t) \geq s$  for all  $s \in S(t)$ .

The second step is to relate the value of problem  $(P_2)$  with inputs  $a(t)$  and  $b(t)$  to the value of problem  $(P_2)$  with inputs  $\hat{\theta}(t)$  and  $\sigma^2(t)$ . We recall that, by definition, we have

$$\begin{aligned} a_i(t) &= \lceil \xi(t) \hat{\theta}_i(t) \rceil, i = 1, \dots, d \\ b_i(t) &= \xi^2(t) \sigma^2(t), i = 1, \dots, d \end{aligned}$$

where  $\xi(t) = \lceil \frac{m}{\delta_t} \rceil$ . Therefore,  $a(t) \in \{1, \dots, \xi(t)\}^d$  and:

$$\hat{\theta}(t) \leq \frac{1}{\xi(t)} a(t) \leq \frac{1}{\xi(t)} \mathbf{1} + \hat{\theta}(t)$$

As a consequence, for any  $x \in \mathcal{X}$ :

$$\hat{\theta}(t)^\top x \leq \frac{1}{\xi(t)} a(t)^\top x \leq \frac{1}{\xi(t)} \mathbf{1}^\top x + \hat{\theta}(t)^\top x \leq \delta_t + \hat{\theta}(t)^\top x$$

We have proven in the first step that:

$$\begin{aligned} & \max_{x \in \mathcal{X}} \left\{ a(t)^\top x + \sqrt{b(t)^\top x} \right\} \\ &\leq a(t)^\top x(t) + \frac{1}{\varepsilon_t} \sqrt{b(t)^\top x(t)} \end{aligned}$$

Then:

$$\begin{aligned}
& \max_{x \in \mathcal{X}} \{\hat{\theta}(t)^\top x + \sqrt{\sigma^2(t)^\top x}\} \\
&= \frac{1}{\xi(t)} \max_{x \in \mathcal{X}} \{\xi(t) \hat{\theta}(t)^\top x + \sqrt{\xi(t)^2 \sigma^2(t)^\top x}\} \\
&\leq \frac{1}{\xi(t)} \max_{x \in \mathcal{X}} \{a(t)^\top x + \sqrt{b(t)^\top x}\} \\
&\leq \frac{1}{\xi(t)} a(t)^\top x(t) + \frac{1}{\xi(t) \varepsilon_t} \sqrt{b(t)^\top x(t)} \\
&\leq \delta_t + \hat{\theta}(t)^\top x(t) + \frac{1}{\varepsilon_t} \sqrt{\sigma^2(t)^\top x(t)}.
\end{aligned}$$

which is the announced result.

## F Pseudo-code to solve problem $(P_2)$

In this section, we provide the pseudo-code for the algorithms described in Section 5, for each family of combinatorial sets. We make use of several subroutines: **Hungarian** is the Hungarian algorithm computing a maximum weighted matching in a bipartite graph, **Greedy** is the greedy algorithm finding a maximum weighted spanning tree of a graph, **Meggido** is Meggido's search algorithm to minimize a piecewise-linear function, **Bellman** is the Bellman-Ford algorithm for the shortest path in a directed acyclic graph.

**input :** Number  $m \in \{1, \dots, d\}$ , rewards  $b \in \mathbb{R}^{|E|}$ , weights  $a \in \mathbb{N}^{|E|}$   
**Result:** Solution to  $P_3(s)$  for all  $s \in \{0, 1, \dots, d \max_i a_i\}$   
 $L \leftarrow$  empty array of size  $(d \max_i a_i + 1, m, d)$   
 $S \leftarrow$  empty array of size  $(d \max_i a_i + 1, m, d)$   
 $S^* \leftarrow$  empty array of size  $(d \max_i a_i + 1)$  **for**  $s = 0, 1, \dots, d \max_i a_i$  **do**  
    **for**  $\ell = 0, 1, \dots, m$  **do**  
        **for**  $i = d, d-1, \dots, 0$  **do**  
            **if**  $i = d$  **then**  
                **if**  $s = 0$  **then**  
                     $L[s, \ell, i] \leftarrow 0$   
                     $S[s, \ell, i] \leftarrow \emptyset$   
                **else**  
                     $L[s, \ell, i] \leftarrow -\infty$   
                     $S[s, \ell, i] \leftarrow \#$   
                **end**  
            **else**  
                **if**  $\ell = 0$  **then**  
                     $L[s, \ell, i] \leftarrow L[s, \ell, i+1]$   
                     $S[s, \ell, i] \leftarrow S[s, \ell, i+1]$   
                **else**  
                     $L[s, \ell, i] \leftarrow \max\{b_i + L[\max(s - a_i, 0), \ell, i+1], L[s, \ell, i+1]\}$  **if**  
                         $L[s, \ell, i] = L[s, \ell, i+1]$  **then**  
                             $S[s, \ell, i] \leftarrow S[s, \ell, i+1]$   
                        **else**  
                             $S[s, \ell, i] \leftarrow S[s, \ell, i+1] \cup \{i\}$   
                        **end**  
                **end**  
            **end**  
        **end**  
    **end**  
     $S^*[s] \leftarrow S[s, m, 0]$   
**end**  
**return**  $S^*$

**Algorithm 1:** Algorithm for the budgeted  $m$ -set problem.

**input :** A directed graph  $G = (V, E)$ , source node  $u \in V$ , destination node  $v \in V$ , length of the longest path  $m$ , rewards  $b \in \mathbb{R}^{|E|}$ , weights  $a \in \mathbb{N}^{|E|}$

**Result:** Solution to  $P_3(s)$  for all  $s \in \{0, 1, \dots, m \max_i a_i\}$

$L \leftarrow$  empty array of size  $(m \max_i a_i + 1, |V|)$

$S \leftarrow$  empty array of size  $(m \max_i a_i + 1, |V|)$

$S^* \leftarrow$  empty array of size  $(d \max_i a_i + 1)$

**for**  $s = 0, 1, \dots, d \max_i a_i$  **do**

**for**  $w \in V$  **do**

**if**  $s = 0$  **then**

$L[s, w], S[s, w] \leftarrow \text{Bellman}(G, b, u, w)$

**else**

$x^* \leftarrow \arg \max_{x: (w, x) \in E} \{b_{(w, x)} + L[x, \max(s - a_{(w, x)}, 0)]\}$

$L[s, w] \leftarrow b_{(w, x^*)} + L[x^*, \max(s - a_{(w, x^*)}, 0)]$

$S[s, w] \leftarrow (w, x) \cup S[x^*, \max(s - a_{(w, x^*)}, 0)]$

**end**

**end**

$S^*[s] \leftarrow S[s, v]$

**end**

**return**  $S^*$

**Algorithm 2:** Algorithm for the budgeted source-destination path problem.

**input :** An undirected graph  $G = (V, E)$ , rewards  $b \in \mathbb{R}^{|E|}$ , weights  $a \in \mathbb{N}^{|E|}$ , a minimum budget  $s \in \mathbb{N}$

**Result:** Solution to  $P_3(s)$

$x \leftarrow \text{nothing}$

**for** all unordered pairs  $(e_1, e_2)$  of distinct edges of  $E$  **do**

$E' \leftarrow \{e \in E \mid b_e \leq \min\{b_{e_1}, b_{e_2}\}\}$

$G' \leftarrow (V, E')$

$x^*, \lambda^* \leftarrow \text{Meggido}(\text{Greedy}, G', a + \lambda b)$

$\varepsilon \leftarrow$  arbitrary small value

$x^+ \leftarrow \text{Greedy}(G', a + (\lambda^* + \varepsilon)b)$

$x^- \leftarrow \text{Greedy}(G', a + (\lambda^* - \varepsilon)b)$

**while**  $|x^+ \oplus x^-| > 1$  **do**

        Find  $e, e'$  such that  $x_e^+ = x_{e'}^- = 1$  and  $x_{e'}^+ = x_e^- = 0$

$\tilde{x} \leftarrow x^+ - e_e + e_{e'}$

**if**  $a^\top \tilde{x} \geq s$  **then**

$x^+ \leftarrow \tilde{x}$

**else**

$x^- \leftarrow \tilde{x}$

**end**

**end**

**if**  $x$  is nothing **or**  $b^\top \tilde{x} > b^\top x$  **then**

$x \leftarrow x^+$

**end**

**end**

**return**  $x$

**Algorithm 3:** Approximation algorithm for the budgeted maximum spanning tree problem.

## References

- [1] Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Proc. of NIPS*. 2011.
- [2] Venkatachalam Anantharam, Pravin Varaiya, and Jean Walrand. Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays-part i: iid rewards. *Automatic Control, IEEE Transactions on*, 32(11):968–976, 1987.
- [3] Alper Atamtürk and Andrés Gómez. Maximizing a class of utility functions over the vertices of a polytope. *Operations Research*, 65(2):433–445, 2017.
- [4] Peter Auer, Nicolò Cesa-Bianchi, Yoav Freund, and Robert E. Schapire. The nonstochastic multiarmed bandit problem. *SIAM J. Comput.*, 32(1):48–77, 2003.
- [5] Andre Berger, Vincenzo Bonifaci, Fabrizio Gandoni, and Guido Shaefer. Budgeted matching and budgeted matroid intersection via the gasoline puzzle. *Mathematical Programming*, 2011.
- [6] Jeff. Bezanson, Alan. Edelman, Stefan. Karpinski, and Viral B. Shah. Julia: A fresh approach to numerical computing. *SIAM Review*, 59(1):65–98, 2017.
- [7] Olivier Cappé, Aurelien Garivier, Odalric Maillard, Remi Munos, and Gilles Stoltz. Kullback-leibler upper confidence bounds for optimal sequential allocation. *Annals of Statistics*, 41(3):516–541, June 2013.
- [8] Nicolò Cesa-Bianchi and Gábor Lugosi. Combinatorial bandits. *Journal of Computer and System Sciences*, 78(5):1404–1422, 2012.
- [9] Wei Chen, Yajun Wang, and Yang Yuan. Combinatorial multi-armed bandit: General framework, results and applications. In *Proc. of ICML*, 2013.
- [10] Wei Chu, Lihong Li, Lev Reyzin, and Robert Schapire. Contextual bandits with linear payoff functions. In *Proc. of AISTATS*, 2011.
- [11] Chris Coey, Miles Lubin, and Juan Pablo Vielma. Outer approximation with conic certificates for mixed-integer convex problems. *Math. Program. Comput.*, 12(2):249–293, 2020.
- [12] Richard Combes, M. Sadegh Talebi, Alexandre Proutiere, and Marc Lelarge. Combinatorial bandits revisited. In *Proc. of NIPS*, 2015.
- [13] Varsha Dani, Thomas P. Hayes, and Sham M. Kakade. Stochastic linear optimization under bandit feedback. In *Proc. of COLT*, pages 355–366, 2008.
- [14] Remy Degenne and Vianney Perchet. Combinatorial semi-bandit with known covariance. In *Proc. of NIPS*, 2016.
- [15] Branislav Kveton, Zheng Wen, Azin Ashkan, Hoda Eydgahi, and Brian Eriksson. Matroid bandits: Fast combinatorial optimization with learning. In *Proc. of UAI*, 2014.
- [16] Branislav Kveton, Zheng Wen, Azin Ashkan, and Csaba Szepesvari. Tight regret bounds for stochastic combinatorial semi-bandits. In *Proc. of AISTATS*, 2015.
- [17] Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4–2, 1985.
- [18] Miguel Sousa Lobo, Lieven Vandenbergh, Stephen Boyd, and Hervé Lebrete. Applications of second-order cone programming. *Linear Algebra and its Applications*, 284(1):193 – 228, 1998.
- [19] Nimrod Megiddo. Applying parallel computation algorithms in the design of serial algorithms. In *Proc. of FOCS*, 1981.

- [20] James G. Oxley. *Matroid Theory (Oxford Graduate Texts in Mathematics)*. Oxford University Press, Inc., USA, 2006.
- [21] Pierre Perrault, Vianney Perchet, and Michal Valko. Exploiting structure of uncertainty for efficient matroid semi-bandits. In *Proc. of ICML*, 2019.
- [22] R. Ravi and Michel X. Goemans. The constrained minimum spanning tree problem. *SWAT*, 1996.
- [23] Idan Rejwan and Yishay Mansour. Top- $k$  combinatorial bandits with full-bandit feedback. *Proc. of ALT*, 2020.
- [24] Herbert Robbins. Some aspects of the sequential design of experiments. *Bulletin of the American Mathematical Society*, 58(5):527–535, 1952.
- [25] M. Sadegh Talebi and Alexandre Proutiere. An optimal algorithm for stochastic matroid bandit optimization. In *Proc. of ICAAMS*, 2016.
- [26] Siwei Wang and Wei Chen. Thompson sampling for combinatorial semi-bandits. In *Proc. of ICML*, 2018.
- [27] Zheng Wen, Branislav Kveton, and Azin Ashkan. Efficient learning in large-scale combinatorial semi-bandits. In *Proc. of ICML*, 2015.

**input :** A bipartite graph  $G = (V, E)$ , rewards  $b \in \mathbb{R}^{|E|}$ , weights  $a \in \mathbb{N}^{|E|}$ , a minimum budget  $s \in \mathbb{N}$   
**Result:** Solution to  $P_3(s)$

$x \leftarrow \text{nothing}$

**for** all unordered 4-tuples  $(e_1, e_2, e_3, e_4)$  of distinct edges of  $E$  **do**

$E' \leftarrow \{e \in E \mid b_e \leq \min\{b_{e_1}, b_{e_2}, b_{e_3}, b_{e_4}\}\}$

$G' \leftarrow (V, E')$

$x^*, \lambda^* \leftarrow \text{Meggido}(\text{Hungarian}, G', a + \lambda b)$

$\varepsilon \leftarrow$  arbitrary small value

$x^+ \leftarrow \text{Hungarian}(G', a + (\lambda^* + \varepsilon)b)$

$x^- \leftarrow \text{Hungarian}(G', a + (\lambda^* - \varepsilon)b)$

**while**  $|x^+ \oplus x^-| > 2$  **do**

$x' \leftarrow x^+ \oplus x^-$

$x'' \leftarrow$  one path or one cycle from  $x'$

$\tilde{x} \leftarrow x^- \oplus x''$

**if**  $a^\top \tilde{x} \geq s$  **then**

$x^+ \leftarrow \tilde{x}$

**else**

$x^- \leftarrow \tilde{x}$

**end**

**end**

**if**  $x$  is nothing **or**  $b^\top \tilde{x} > b^\top x$  **then**

$x \leftarrow x^+$

**end**

**end**

**return**  $x$

**Algorithm 4:** Approximation algorithm for the budgeted maximum bipartite matching problem.