Exploiting Structure of Uncertainty for Efficient Matroid Semi-Bandits

Pierre Perrault [1 2] **Vianney Perchet** [2 3] **Michal Valko** [1 4]

Abstract

We improve the efficiency of algorithms for stochastic *combinatorial semi-bandits*. In most interesting problems, state-of-the-art algorithms take advantage of structural properties of rewards, such as *independence*. However, while being optimal in terms of asymptotic regret, these algorithms are inefficient. In our paper, we first reduce their implementation to a specific *submodular maximization*. Then, in case of *matroid* constraints, we design adapted approximation routines, thereby providing the first efficient algorithms that rely on reward structure to improve regret bound. In particular, we improve the state-of-the-art efficient gap-free regret bound by a factor $\sqrt{m}/\log m$, where m is the maximum action size. Finally, we show how our improvement translates to more general *budgeted combinatorial semi-bandits*.

1. Introduction

Stochastic bandits model sequential decision-making in which an *agent* selects an arm (a decision) at each round and observes a realization of the corresponding unknown reward distribution. The goal is to maximize the expected cumulative reward, or equivalently, to minimize the *expected regret*, defined as the difference between the expected cumulative reward achieved by an oracle algorithm always selecting the optimal arm and that achieved by the agent. To accomplish this objective, the agent must trade-off between *exploration* (gaining information about reward distributions) and *exploitation* (using greedily the information collected so far) as it was already discovered by Robbins (1952). Bandits have been applied to many fields such as mechanism design (Mohri & Munoz, 2014), search advertising (Tran-Thanh et al., 2014), and personalized recommendation (Li et al., 2010). We improve the computational efficiency (i.e., the time and space complexity) of their *combinatorial* generalization, in which the agent selects at each round a *subset* of arms, that we refer to as an *action* in the rest of the paper (Cesa-Bianchi & Lugosi, 2012; Gai et al., 2012; Audibert et al., 2014; Chen et al., 2014).

Different kinds of feedback provided by the environment are possible. First, with *bandit feedback* (also called full bandit or opaque feedback), the agent only observes the *total* reward associated to the selected action. Second, with *semi-bandit feedback*, the agent observes the *partial* reward of each base arm in the selected action. Finally, with *full information feedback*, the agent observes the partial reward of all arms. We give results for semi-bandit feedback only.

There are two main questions that come up with combinatorial (semi)-bandits: 1° *How can the stochastic structure of the reward vector be exploited to reduce the regret?* and 2° *Can algorithms be efficient?* Combes et al. (2015) answer the first question assuming that reward distributions are *mutually independent*. Later, Degenne & Perchet (2016) generalize the algorithm of Combes et al. (2015) to a larger class of *sub-Gaussian* rewards by exploiting the covariance structure of the arms. They also show the optimality of proposed algorithms, in particular, that an upper bound on their regret matches the asymptotic gap dependent lower bound of this class. However, algorithms of Combes et al. (2015) and Degenne & Perchet (2016) are computationally inefficient. The second question is studied by Kveton et al. (2015), who give an efficient algorithm based on the UCB algorithm of Auer et al. (2002). While being efficient, the algorithm of Kveton et al. (2015) assumes the worst case class of *arbitrary correlated*[1] rewards, i.e., it does not exploit any properties of rewards and therefore does not match the lower bound of Degenne & Perchet (2016).

On the other hand, efficient algorithms for matroid (Whitney, 1935) semi-bandits exist (Kveton et al., 2014; Talebi & Proutiere, 2016) and their regret bounds match the asymptotic gap dependent lower bound, which is the same for both sub-Gaussian and arbitrary correlated rewards. Among these algorithms, the state-of-the-art gap-free regret bound is of order $\mathcal{O}\big(\sqrt{nmT\log T}\big)$, where T is the number of rounds, n is the number of base arms, and m is the maximum action size (Kveton et al., 2014).

[1]SequeL team, INRIA Lille - Nord Europe [2]CMLA, ENS Paris-Saclay [3]Criteo AI Lab [4]now with DeepMind. Correspondence to: P. Perrault <pierre.perrault@inria.fr>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

[1]Any dependence can exist between rewards.

Our contributions In this paper, we show how algorithms of Combes et al. (2015) and Degenne & Perchet (2016) can be *efficiently* approximated for matroid. This improves the bound of Kveton et al. (2014) by a factor $\sqrt{m}/\log(m)$ for the class of *sub-Gaussian* rewards. We first locate the source of inefficiency of these algorithms: At each round, they have to solve a *submodular maximization* problem. We then provide efficient, adapted LocalSearch and Greedy-based algorithms that exploit the submodularity to give approximation guarantees on the regret upper bound. These algorithms can be of independent interest. We also extend our approximation techniques to more challenging *budgeted combinatorial semi-bandits* via binary search methods and exhibit the same improvement for this setting as well.

Related work Efficiency in combinatorial bandits, or more generally, in linear bandits with a large set of arms is an open problem. Some methods, such as *convex hull representation*, (Koolen et al., 2010), *hashing* (Jun et al., 2017), or *DAG-encoding* (Sakaue et al., 2018) reduce the algorithmic complexity. For semi-bandit feedback, in the adversarial case, Neu & Bartók (2013) proposed an efficient implementation via *geometric resampling*. In the stochastic case, many efficient Bayesian algorithms exist (see e.g., Russo & Van Roy, 2013; Russo & Roy, 2016), although they are only shown to be optimal for *Bayesian regret*.[2]

2. Background

We denote the set of arms by $[n] \triangleq \{1, 2, \dots, n\}$, we typeset vectors in bold and indicate components with indices, i.e., $\mathbf{a} = (a_i)_{i\in[n]} \in \mathbb{R}^n$. We let $\mathcal{P}([n]) \triangleq \{A, A \subset [n]\}$ be the power set of $[n]$. Let $\mathbf{e}_i \in \mathbb{R}^n$ denote the i^{th} canonical unit vector. The *incidence vector* of any set $A \in \mathcal{P}([n])$ is

$$\mathbf{e}_A \triangleq \sum_{i\in A} \mathbf{e}_i.$$

The above definition allows us to represent a subset of $[n]$ as an element of $\{0,1\}^n$. We denote the Minkowski sum of two sets $Z, Z' \subset \mathbb{R}^n$ as $Z + Z' \triangleq \{z + z', z \in Z, z' \in Z'\}$, and $Z + z' \triangleq Z + \{z'\}$. Let $\mathcal{A} \subset \mathcal{P}([n])$ be a set of *actions*. We define the maximum possible cardinality of an element of $\mathcal{A}$ as $m \triangleq \max\{|A|,\ A \in \mathcal{A}\}$.

2.1. Stochastic Combinatorial Semi-Bandits

In combinatorial semi-bandits, an agent selects an action $A_t \in \mathcal{A}$ at each round $t \in \mathbb{N}^\star$, and receives a reward $\mathbf{e}_{A_t}^\intercal \mathbf{X}_t$, where $\mathbf{X}_t \in \mathbb{R}^n$ is an unknown random vector of rewards. The successive reward vectors $(\mathbf{X}_t)_{t\geq 1}$ are i.i.d., with an unknown mean $\boldsymbol{\mu}^\star \triangleq \mathbb{E}[\mathbf{X}] \in \mathbb{R}^n$, where $\mathbf{X} = \mathbf{X}_1$. After selecting an action A_t in round t, the agent observes the partial reward of each individual arm in A_t. The goal of the agent is to minimize the expected regret, defined with $A^\star \in \arg\max_{A\in\mathcal{A}} \mathbf{e}_A^\intercal \boldsymbol{\mu}^\star$ as

$$R_T \triangleq \mathbb{E}\left[\sum_{t=1}^{T} (\mathbf{e}_{A^\star} - \mathbf{e}_{A_t})^\intercal \mathbf{X}_t\right].$$

For any action $A \in \mathcal{A}$, we define its gap as the difference $\Delta(A) \triangleq (\mathbf{e}_{A^\star} - \mathbf{e}_A)^\intercal \boldsymbol{\mu}^\star$. We then rewrite the expected cumulative regret as $R_T = \mathbb{E}\left[\sum_{t=1}^T \Delta(A_t)\right]$. Finally, we define $\Delta \triangleq \min_{A\in\mathcal{A},\ \Delta(A)>0} \Delta(A)$.

Combinatorial semi-bandits have been introduced by Cesa-Bianchi & Lugosi (2012). More recently, different algorithms have been proposed (Talebi et al., 2013; Combes et al., 2015; Kveton et al., 2015; Degenne & Perchet, 2016), depending whether the random vector $\mathbf{X}$ satisfies specific properties. Some of these properties commonly assumed are a subset of the following ones:

(i) $X_1, \dots, X_n \in \mathbb{R}$ are *mutually independent*,

(ii) $X_1, \dots, X_n \in \mathbb{R}$ are *arbitrary correlated*,

(iii) $\mathbf{X} \in [-1, 1]^n$,

(iv) $\mathbf{X} \in \mathbb{R}^n$ is *multivariate sub-Gaussian*, i.e., $\mathbb{E}\left[e^{\boldsymbol{\lambda}^\intercal(\mathbf{X}-\boldsymbol{\mu}^\star)}\right] \leq e^{\|\boldsymbol{\lambda}\|_2^2/2},\ \forall \boldsymbol{\lambda} \in \mathbb{R}^n$,

(v) $\mathbf{X} \in \mathbb{R}^n$ is *component-wise sub-Gaussian*, i.e., $\mathbb{E}\left[e^{\lambda_i(X_i-\mu_i^\star)}\right] \leq e^{\lambda_i^2/2},\ \forall i \in [n],\ \forall \boldsymbol{\lambda} \in \mathbb{R}^n$.

2.2. Lower Bounds

Combining some of the above properties, we consider different classes of possible distributions for $\mathbf{X}$. In Table 1, we show two existing gap-dependent lower bounds on R_T that depend on the respective class. They are valid for at least one distribution of $\mathbf{X}$ belonging to the corresponding class, one combinatorial structure $\mathcal{A} \subset \mathcal{P}([n])$, and for any consistent algorithm (Lai & Robbins, 1985), for which the regret on any problem verifies $R_T = o(T^a)$ as $T \to \infty$ for all $a > 0$. Table 1 suggests that a tighter regret rate can be reached with some prior knowledge on the random vector $\mathbf{X}$.

3. (In)efficiency of Existing Algorithms

In this section, we discuss the efficiency of existing algorithms matching the lower bounds in Table 1. We consider that an algorithm is *efficient* as soon as the time and space complexity for each round t is polynomial in n and polylog-

[2] A setting where arm distributions depend on a random parameter drawn from a known prior, and expectation of the regret is also taken with respect to this parameter.

Table 1. Gap-dependent lower bounds proved on different classes of possible distributions for $\mathbf{X}$.

Class of possible reward distributions	*Gap-dependent lower bound*
$(i)+(iii)$ $\Rightarrow(i)+(v)$ $\Rightarrow(iv)$	$\Omega\left(\frac{n\log T}{\Delta}\right)$ Combes et al., 2015
$(ii)+(iii)$ $\Rightarrow(ii)+(v)$	$\Omega\left(\frac{nm\log T}{\Delta}\right)$ Kveton et al., 2015

arithmic[3] in t. Notice that the per-round complexity depends substantially on $\mathcal{A}$. We assume $\mathcal{A}$ is such that linear optimization problems on $\mathcal{A}$ — of the form $\max_{A\in\mathcal{A}} \mathbf{e}_A^\intercal\boldsymbol{\delta}$ for some $\boldsymbol{\delta}\in\mathbb{R}^n$ — can be solved efficiently. As a consequence, an agent knowing $\boldsymbol{\mu}^\star$ can efficiently compute $A^\star$. Assuming efficient linear maximization is crucial (cf. Neu & Bartók, 2013; Combes et al., 2015; Kveton et al., 2015; Degenne & Perchet, 2016). Without this assumption, e.g., for $\mathcal{A}$ being dominating sets in a graph, even the offline problem cannot be solved efficiently, and we would have to consider the notion of approximation regret instead, as was done by Chen et al. (2013).

3.1. A Generic Algorithm

As mentioned above, the action A_t is selected based on the feedback received up to round $t-1$. A common statistic computed from this feedback is the *empirical average* of each arm $i\in[n]$, defined as

$$\overline{\mu}_{i,0}=0,\quad \forall t\geq 2,\ \overline{\mu}_{i,t-1}=\frac{\sum_{u\in[t-1]}\mathbb{I}\{i\in A_u\}X_{i,u}}{N_{i,t-1}},$$

where $\forall t\geq 1,\ N_{i,t-1}\triangleq\sum_{u\in[t-1]}\mathbb{I}\{i\in A_u\}$. Many combinatorial semi-bandit algorithms, in particular, those listed in Table 2, can be seen as a special case of Algorithm 1 for different confidence regions $\mathcal{C}_t$ around $\overline{\boldsymbol{\mu}}_{t-1}$.

Algorithm 1 Generic confidence-region-based algorithm.

At each round t :
  Find a confidence region $\mathcal{C}_t\subset\mathbb{R}^n$.
  Solve the bilinear program

$$(\boldsymbol{\mu}_t,A_t)\in\operatorname*{arg\,max}_{\boldsymbol{\mu}\in\mathcal{C}_t,A\in\mathcal{A}}\mathbf{e}_A^\intercal\boldsymbol{\mu}\ .$$

  Play A_t.

We further assume that $\mathcal{C}_t$ is defined through some parameters $p,r\in\{1,\infty\}$, and some functions $g_{i,t}$, $i\in[n]$ by

$$\mathcal{C}_t\triangleq[-r,r]^n\cap\left(\overline{\boldsymbol{\mu}}_{t-1}+\left\{\boldsymbol{\delta}\in\mathbb{R}^n,\left\|(g_{i,t}(\delta_i))_i\right\|_p\leq 1\right\}\right),$$

where $g_{i,t}=0$ if $N_{i,t-1}=0$ and, otherwise, is convex, strictly decreasing on $[-r-\overline{\mu}_{i,t-1},0]$ and strictly increasing on $[0,r-\overline{\mu}_{i,t-1}]$ such that $g_{i,t}(0)=0$. Typically, $r=1$ under assumption (iii) and $r=\infty$ otherwise. Table 2 lists variants of Algorithm 1, with the corresponding reward class under which they can be used. Each of these algorithms is matching the lower bound corresponding to the respective reward class considered in Table 1, i.e., R_T is a 'big $\mathcal{O}$' of the lower bound, up to a polylogarithmic factor in m (Degenne & Perchet, 2016). Notice that THOMPSONSAMPLING is not an instance of Algorithm 1. However, we are not aware of any tight analysis: The one by Wang & Chen (2018) matches the lower bound $nm\log(T)/\Delta$, but only for mutually independent rewards, where the lower bound is $n\log(T)/\Delta$. The regret upper bound of algorithms in Table 1 with $p=1$ have an additive constant term w.r.t. T but exponential in n, which can be replaced with a different analysis to get either:

- an exponential term in m plus a term of order $1/\Delta^2$,
- a term of order $1/\Delta^2$ — by changing $\log(t)$ to $\log(t)+m\log\log(t)$ in the algorithm,
- or can be removed — by changing $\log(t)$ to $\log(t)+n\log\log(t)$ in the algorithm.

On one hand, the arbitrary correlated case can be considered as solved, since the matching lower bound algorithm CUCB (Kveton et al., 2015) is efficient.[4] On the other hand, considering the reward class given by the first line of Table 1, the known algorithms that match the lower bound are inefficient.[5] We further discuss the efficiency of Algorithm 1 in the following subsection.

3.2. Submodular Maximization

In Algorithm 1, only A_t needs to be computed. It is a maximizer over $\mathcal{A}$ of the set function

$$\begin{array}{ccc}\mathcal{P}([n]) & \rightarrow & \mathbb{R}\\ A & \mapsto & \max_{\boldsymbol{\mu}\in\mathcal{C}_t}\mathbf{e}_A^\intercal\boldsymbol{\mu}\ .\end{array}\tag{1}$$

We can easily evaluate the function (1) above for some set $A\in\mathcal{P}([n])$, since it only requires solving a linear optimization problem on the convex[6] set $\mathcal{C}_t$. In Proposition 1, we

[3] In streaming settings with near real-time requirements, it is imperative to have algorithms that can run with a complexity that stay almost constant across rounds.

[4] In Theorem 1, we recover that A_t is computed by Algorithm 1 by solving a linear optimization problem.

[5] $\mathcal{A}$ may have up to 2^n elements.

[6] $\mathcal{C}_t$ is convex since functions $g_{i,t}$ are convex.

Table 2. Some combinatorial semi-bandit algorithms and their properties.

Class	*Algorithm*	p	$g_{i,t}$ (*for* $N_{i,t-1} \geq 1$, *up to universal factor*)	*Efficient?*
$(ii) + (v)$	CUCB (Kveton et al., 2015)	∞	$\delta \mapsto \dfrac{\delta^2 N_{i,t-1}}{\log(t)}$	Yes
$(i) + (iii)$	ESCB-KL (Combes et al., 2015)	1	$\delta \mapsto \dfrac{\mathrm{kl}\left(\frac{1+\overline{\mu}_{i,t-1}}{2}, \frac{1+\overline{\mu}_{i,t-1}+\delta}{2}\right) N_{i,t-1}}{\log(t)}$	No
(iv)	ESCB (Combes et al., 2015), OLS-UCB (Degenne & Perchet, 2016)	1	$\delta \mapsto \dfrac{\delta^2 N_{i,t-1}}{\log(t)}$	No

show that in some cases, the evaluation can be even simpler. However, maximizing the function (1) over a combinatorial set $\mathcal{A}$ is not straightforward. Before studying this function more closely, Definition 1 recalls some well-known properties that can be satisfied by a set function $F : \mathcal{P}([n]) \to \mathbb{R}$.

Definition 1. *A set function F is:*

- *normalized, if $F(\emptyset) = 0$,*
- *linear (or modular) if $F(A) = \mathbf{e}_A^\mathsf{T}\boldsymbol{\delta} + b$, for some $\boldsymbol{\delta} \in \mathbb{R}^n$, $b \in \mathbb{R}$,*
- *non-decreasing if $F(A) \leq F(B)\ \forall\ A \subset B \subset [n]$,*
- *submodular if for all $A, B \subset [n]$,*

$$F(A \cup B) + F(A \cap B) \leq F(A) + F(B).$$

Equivalently, F is submodular if for all $A \subset B \subset [n]$, and $i \notin B$, $F(A\cup\{i\}) - F(A) \geq F(B\cup\{i\}) - F(B)$.

The function (1) is clearly normalized, and it can be decomposed into two set functions in the following way,

$$\forall A \subset [n],\ \max_{\boldsymbol{\mu}\in\mathcal{C}_t} \mathbf{e}_A^\mathsf{T}\boldsymbol{\mu} = \mathbf{e}_A^\mathsf{T}\overline{\boldsymbol{\mu}}_{t-1} + \max_{\boldsymbol{\delta}\in\mathcal{C}_t-\overline{\boldsymbol{\mu}}_{t-1}} \mathbf{e}_A^\mathsf{T}\boldsymbol{\delta}.$$

The linear part $A \mapsto \mathbf{e}_A^\mathsf{T}\overline{\boldsymbol{\mu}}_{t-1}$ is efficiently maximized alone, we thus focus on the other part, $A \mapsto \max_{\boldsymbol{\delta}\in\mathcal{C}_t-\overline{\boldsymbol{\mu}}_{t-1}} \mathbf{e}_A^\mathsf{T}\boldsymbol{\delta}$, usually called an *exploration bonus*. It aims to compensate for the negative selection bias of the first term. We define

$$\begin{aligned}\mathcal{C}_t^+ &\triangleq [-r, r]^n \cap \left(\overline{\boldsymbol{\mu}}_{t-1} + \left\{\boldsymbol{\delta} \in \mathbb{R}_+^n, \left\|(g_{i,t}(\delta_i))_i\right\|_p \leq 1\right\}\right) \\ &= \mathcal{C}_t \cap \left\{\boldsymbol{\mu} \in \mathbb{R}^n,\ \boldsymbol{\mu} \geq \overline{\boldsymbol{\mu}}_{t-1}\right\}\end{aligned}$$

and rewrite $A \mapsto \max_{\boldsymbol{\delta}\in\mathcal{C}_t-\overline{\boldsymbol{\mu}}_{t-1}} \mathbf{e}_A^\mathsf{T}\boldsymbol{\delta}$ through Lemma 1.

Lemma 1. *For all $A \in \mathcal{P}([n])$, $\max_{\boldsymbol{\delta}\in\mathcal{C}_t-\overline{\boldsymbol{\mu}}_{t-1}} \mathbf{e}_A^\mathsf{T}\boldsymbol{\delta} = \max_{\boldsymbol{\delta}\in\mathcal{C}_t^+-\overline{\boldsymbol{\mu}}_{t-1}} \mathbf{e}_A^\mathsf{T}\boldsymbol{\delta}$.*

The lemma holds as $\left\{\left(\delta_i^+\right)_i,\ \boldsymbol{\delta} \in \mathcal{C}_t - \overline{\boldsymbol{\mu}}_{t-1}\right\} \subset \mathcal{C}_t - \overline{\boldsymbol{\mu}}_{t-1}$. As a corollary, this set function is non-negative, and non-decreasing. It can be written in closed form under additional assumptions, see Proposition 1 and Example 1.

Proposition 1. *Let $A \in \mathcal{P}([n])$, $t \in \mathbb{N}^\star$, $p = 1$. Assume that for all $i \in A$, $g_{i,t}$ has a strictly increasing, continuous derivative $g_{i,t}'$ defined on $[0, r - \overline{\mu}_{i,t-1}]$. For $i \in A$, let*

$$f_i(\lambda) \triangleq \begin{cases} g_{i,t}'^{-1}(1/\lambda) & \text{if } 1/\lambda < g_{i,t}'(r - \overline{\mu}_{i,t-1}), \\ r - \overline{\mu}_{i,t-1} & \text{otherwise,} \end{cases}$$

defined for $\lambda \geq 0$. Then, the smallest $\lambda^\star$ satisfying

$$\mathbf{e}_A^\mathsf{T}(g_{i,t}(f_i(\lambda^\star)))_i \leq 1$$

is such that

$$(\delta_i^\star)_i \triangleq (\mathbb{I}\{i \in A\} f_i(\lambda^\star))_i \in \operatorname*{arg\,max}_{\boldsymbol{\delta}\in\mathcal{C}_t^+-\overline{\boldsymbol{\mu}}_{t-1}} \mathbf{e}_A^\mathsf{T}\boldsymbol{\delta}.$$

The proof of Proposition 1 can be found in Appendix A. An important use-case example of Proposition 1 is the following

Example 1. *Let $A \in \mathcal{P}([n])$, $t \in \mathbb{N}^\star$. If for all $i \in [n]$, $g_{i,t} = (\cdot)^2\alpha_{i,t}$ for some $\alpha_{i,t} > 0$, and $r = \infty, p = 1$, then*

$$\max_{\boldsymbol{\delta}\in\mathcal{C}_t^+-\overline{\boldsymbol{\mu}}_{t-1}} \mathbf{e}_A^\mathsf{T}\boldsymbol{\delta} = \sqrt{\mathbf{e}_A^\mathsf{T}\left(\frac{1}{\alpha_{i,t}}\right)_i}.$$

Indeed, since the maximizer $\boldsymbol{\delta}^\star$ lies at the boundary, $\max_{\boldsymbol{\delta}\in\mathcal{C}_t^+-\overline{\boldsymbol{\mu}}_{t-1}} \mathbf{e}_A^\mathsf{T}\boldsymbol{\delta} = \max_{\boldsymbol{\delta}\in\mathbb{R}_+^n,\ \sum_i \alpha_{i,t}\delta_i^2=1} \mathbf{e}_A^\mathsf{T}\boldsymbol{\delta}$, and from the first-order optimality condition we deduce that $\mathbf{e}_A = 2\lambda^\star(\alpha_{i,t}\delta_i^\star)_i$, i.e., $\delta_i^\star = \mathbb{I}\{i \in A\}/2\lambda^\star\alpha_{i,t}$, where $\lambda^\star$ is necessarily $\frac{1}{2}\sqrt{\mathbf{e}_A^\mathsf{T}(1/\alpha_{i,t})_i}$. We thus recover the ESCB*'s exploration bonus for $\alpha_{i,t} = N_{i,t-1}/\log t$.*

Remark 1. *The proof of Proposition 1 follows the same technique as the proof of Theorem 4 by Combes et al. (2015) for developing the computation of the* ESCB-KL *exploration bonus.*

Example 1 is a specific case where the exploration bonus $A \mapsto \max_{\boldsymbol{\delta}\in\mathcal{C}_t^+-\overline{\boldsymbol{\mu}}_{t-1}} \mathbf{e}_A^\mathsf{T}\boldsymbol{\delta}$ has a particularly simple form: It is the square root of a non-decreasing linear set function. Such a set function is known to be submodular (Stobbe & Krause, 2010). This interesting property helps for maximizing the function (1). In Theorem 1, we prove that $A \mapsto \max_{\boldsymbol{\delta}\in\mathcal{C}_t^+-\overline{\boldsymbol{\mu}}_{t-1}} \mathbf{e}_A^\mathsf{T}\boldsymbol{\delta}$ is in fact always submodular.

Theorem 1. *The following two properties hold.*

- *For* $p = \infty$, $A \mapsto \max_{\boldsymbol{\delta} \in \mathcal{C}_t^+ - \overline{\boldsymbol{\mu}}_{t-1}} \mathbf{e}_A^\mathsf{T} \boldsymbol{\delta}$ *is linear.*
- *For* $p = 1$, $A \mapsto \max_{\boldsymbol{\delta} \in \mathcal{C}_t^+ - \overline{\boldsymbol{\mu}}_{t-1}} \mathbf{e}_A^\mathsf{T} \boldsymbol{\delta}$ *is submodular.*

The proof is deferred to Appendix B and uses a result on *polymatroids* by He et al. (2012). Theorem 1 first implies the efficiency of any variant of Algorithm 1 with $p = \infty$, since it reduces to optimizing a linear set function over $\mathcal{A}$. Theorem 1 also yields that when the reward class is strengthen to target the tighter lower bound $n \log(T)/\Delta$, Algorithm 1 reduces to maximizing a submodular set function over $\mathcal{A}$ (the sum of a linear and a submodular function is submodular). Submodular maximization problems have been applied in machine learning before (see e.g., Krause & Golovin, 2011; Bach, 2011), however, maximizing a submodular function F, even for $\mathcal{A} = \{A, |A| \leq m\}$ and F non-decreasing, is NP-Hard in general (Schrijver, 2008), with an approximation factor of $1 + 1/(e-1)$ by the GREEDY algorithm (Nemhauser et al., 1978). This is problematic as the typical analysis is based on controlling with high probability the error $\Delta(A_t)$ at round t using $\max_{\boldsymbol{\delta} \in \mathcal{C}_t - \overline{\boldsymbol{\mu}}_{t-1}} \mathbf{e}_{A_t}^\mathsf{T} (|\delta_i|)_i$, which converges to zero as $\mathcal{C}_t - \overline{\boldsymbol{\mu}}_{t-1}$ becomes increasingly tight along axis $i \in A_t$ (because counters $N_{i,t-1}$ increases for $i \in A_t$). More precisely, since $\boldsymbol{\mu}^\star$ belongs with high probability to the confidence region $\mathcal{C}_t$, $\boldsymbol{\mu}^\star - \overline{\boldsymbol{\mu}}_{t-1}$ belongs with high probability to $\mathcal{C}_t - \overline{\boldsymbol{\mu}}_{t-1}$. For $\kappa \geq 1$, a $\kappa-$approximation algorithm for maximizing the function (1) would only guarantee the following:

$$
\begin{aligned}
\Delta(A_t) &= (\mathbf{e}_{A^\star} - \mathbf{e}_{A_t})^\mathsf{T} \boldsymbol{\mu}^\star \\
&\leq \max_{\boldsymbol{\mu} \in \mathcal{C}_t} \mathbf{e}_{A^\star}^\mathsf{T} \boldsymbol{\mu} - \mathbf{e}_{A_t}^\mathsf{T} \boldsymbol{\mu}^\star \\
&= \max_{\boldsymbol{\mu} \in \mathcal{C}_t^+} \mathbf{e}_{A^\star}^\mathsf{T} \boldsymbol{\mu} - \mathbf{e}_{A_t}^\mathsf{T} \boldsymbol{\mu}^\star \\
&\leq \kappa \max_{\boldsymbol{\mu} \in \mathcal{C}_t^+} \mathbf{e}_{A_t}^\mathsf{T} \boldsymbol{\mu} - \mathbf{e}_{A_t}^\mathsf{T} \boldsymbol{\mu}^\star \qquad (2) \\
&= \kappa \max_{\boldsymbol{\delta} \in \mathcal{C}_t^+ - \overline{\boldsymbol{\mu}}_{t-1}} \mathbf{e}_{A_t}^\mathsf{T} \boldsymbol{\delta} \\
&\quad + \mathbf{e}_{A_t}^\mathsf{T} (\overline{\boldsymbol{\mu}}_{t-1} - \boldsymbol{\mu}^\star) + (\kappa - 1) \mathbf{e}_{A_t}^\mathsf{T} \overline{\boldsymbol{\mu}}_{t-1} \qquad (3) \\
&\leq (\kappa + 1) \max_{\boldsymbol{\delta} \in \mathcal{C}_t - \overline{\boldsymbol{\mu}}_{t-1}} \mathbf{e}_{A_t}^\mathsf{T} (|\delta_i|)_i + (\kappa - 1) \mathbf{e}_{A_t}^\mathsf{T} \overline{\boldsymbol{\mu}}_{t-1}.
\end{aligned}
$$

If $\kappa \neq 1$, the term $(\kappa - 1)\mathbf{e}_{A_t}^\mathsf{T} \overline{\boldsymbol{\mu}}_{t-1}$ gives linear regret bounds. In the next section, with a stronger assumption on $\mathcal{A}$ (but for which submodular maximization is still NP-Hard), we show that both parts of the objective can have different approximation factors. More precisely, we show how to approximate the linear part with factor 1, and the submodular part with a constant factor $\kappa > 1$. Then, (2) can be replaced by

$$\kappa \cdot \max_{\boldsymbol{\mu} \in \mathcal{C}_t^+ - \overline{\boldsymbol{\mu}}_{t-1}} \mathbf{e}_{A_t}^\mathsf{T} \boldsymbol{\mu} + 1 \cdot \mathbf{e}_{A_t}^\mathsf{T} \overline{\boldsymbol{\mu}}_{t-1} - \mathbf{e}_{A_t}^\mathsf{T} \boldsymbol{\mu}^\star.$$

Therefore, in (3), the extra $(\kappa - 1)\mathbf{e}_{A_t}^\mathsf{T} \overline{\boldsymbol{\mu}}_{t-1}$ term is removed.

4. Efficient Algorithms for Matroid Constraints

In this section, we will consider additional structure on $\mathcal{A}$, using the notion of matroid, recalled below.

Definition 2. *A matroid is a pair* $([n], \mathcal{I})$*, where* $\mathcal{I}$ *is a family of subsets of* $[n]$*, called the independent sets, with the following properties:*

- *The empty set is independent, i.e.,* $\emptyset \in \mathcal{I}$.
- *Every subset of an independent set is independent, i.e., for all* $A \in \mathcal{I}$*, if* $A' \subset A$*, then* $A' \in \mathcal{I}$.
- *If* A *and* B *are two independent sets, and* $|A| > |B|$*, then there exists* $x \in A \backslash B$ *such that* $B \cup \{x\} \in \mathcal{I}$.

Matroids generalize the notion of linear independence. A maximal (for the inclusion) independent set is called *basis* and all bases have the same cardinality m, which is called the *rank* of the matroid (Whitney, 1935). Many combinatorial problems such as building a spanning tree for network routing (Oliveira & Pardalos, 2005) can be expressed as a linear optimization on a matroid (see Edmonds & Fulkerson, 1965 or Perfect, 1968, for other examples).

Let $\mathcal{I} \in \mathcal{P}([n])$ be such that $([n], \mathcal{I})$ forms a matroid. Let $\mathcal{B} \subset \mathcal{I}$ be the set of bases of the matroid $([n], \mathcal{I})$. In the following, we may assume that $\mathcal{A}$ is either $\mathcal{I}$ or $\mathcal{B}$. We also assume that an independence oracle is available, i.e., given an input $A \subset [n]$, it returns TRUE if $A \in \mathcal{I}$ and FALSE otherwise. Maximizing a linear set function L on $\mathcal{A}$ is efficient, and it can be done as follows (Edmonds, 1971): Let σ be a permutation of $[n]$ and j an integer such that $j = m$ in case $\mathcal{A} = \mathcal{B}$ and otherwise, j satisfies

$$\ell_1 \geq \cdots \geq \ell_j \geq 0 \geq \ell_{j+1} \geq \cdots \geq \ell_n,$$

where $\ell_i = L(\{\sigma(i)\}) \; \forall i \in [n]$. The optimal S is built greedily starting from $S = \emptyset$, and for $i \in [j]$, $\sigma(i)$ is added to S only if $S \cup \{\sigma(i)\} \in \mathcal{I}$.

Matroid bandits with $\mathcal{A} = \mathcal{B}$ has been studied by Kveton et al. (2014); Talebi & Proutiere (2016). In this case, the two lower bounds in Table 1 coincide to $\Omega(n \log(T)/\Delta)$, and CUCB reaches it, with the following gap-free upper bound: $R_T(\text{CUCB}) = \mathcal{O}\left(\sqrt{nmT \log T}\right)$. Assuming sub-Gaussian rewards to use any Algorithm of Table 2 with $p = 1$ would tighten (Degenne & Perchet, 2016) this gap-free upper bound to $\mathcal{O}\left(\sqrt{n \log^2 mT \log T}\right)$. Notice, due to the $\sqrt{\log T}$ factor, this does not contradict the $\Omega\left(\sqrt{nmT}\right)$ gap-free lower bound for multi-play bandits.

In the rest of this section, we provide efficient approximation routines to maximize the function (1) on $\mathcal{A} = \mathcal{I}$ and $\mathcal{B}$ without having the extra undesirable term $(\kappa - 1)\mathbf{e}_{A_t}^\mathsf{T} \overline{\boldsymbol{\mu}}_{t-1}$, that a standard $\kappa-$approximation algorithm would suffer.

Therefore, using these routines in Algorithm 1 do not alter its regret upper bound.

Let L be a normalized, linear set function, that will correspond to the linear part $A \mapsto \mathbf{e}_A^\intercal \overline{\boldsymbol{\mu}}_{t-1}$; and let F denote a normalized, non-decreasing, submodular function, that will correspond to the submodular part $A \mapsto \max_{\boldsymbol{\delta}\in\mathcal{C}_t^+ - \overline{\boldsymbol{\mu}}_{t-1}} \mathbf{e}_A^\intercal \boldsymbol{\delta}$. Unless stated otherwise, we further assume that F is positive (for $A \neq \emptyset$). This is a mild assumption as it holds for $A \mapsto \max_{\boldsymbol{\delta}\in\mathcal{C}_t^+ - \overline{\boldsymbol{\mu}}_{t-1}} \mathbf{e}_A^\intercal \boldsymbol{\delta}$ in the unbounded case, i.e., if (iii) is not assumed and $r = \infty$. If (iii) is true, then adding an extra $\mathbf{e}_A^\intercal \left(\frac{1}{N_{i,t-1}^2}\right)_i$ term will recover positivity and increase the regret upper bound by only an additive constant. In the following subsections, we will provide algorithms that efficiently outputs S such that

$$L(S) + \kappa F(S) \geq L(O) + F(O), \quad \forall O \in \mathcal{A}, \tag{4}$$

with some appropriate approximation factor $\kappa \geq 1$. It is possible to efficiently output S_1 and S_2 such that we get $L(S_1) \geq L(O_1)$ and $\kappa F(S_2) \geq F(O_2)$ for any $O_1, O_2 \in \mathcal{A}$. Although we can take $O_1 = O_2$, S_1 and S_2 are not necessarily equal, and (4) is not straightforward. The last subsection is we apply this approach to *budgeted matroid semi-bandits*.

4.1. Local Search Algorithm

In this subsection, we assume that $\mathcal{A} = \mathcal{I}$. In Algorithm 2, we provide a specific instance of LocalSearch that we tailored to our needs to approximately maximize $L + F$.

It starts from the greedy solution $S_{\text{init}} \in \arg\max_{A\in\mathcal{I}} L(A)$. Then, Algorithm 2 repeatedly tries three basic operations in order to improve the current solution. Since every $S \in \mathcal{A}$ can potentially be visited, only *significative* improvements are considered, i.e., improvements greater than $\frac{\varepsilon}{m}F(S)$ for some input parameter $\varepsilon > 0$. The smaller ε is, the higher complexity will be. Notice the improvement threshold $\frac{\varepsilon}{m}F(S)$ does not depend on L. In fact, this is crucial to ensure that the approximation factor of L is 1. However, this can increase the time complexity. To prevent this increase, the second essential ingredient is the initialization, where only L plays a role.

In Theorem 2, we state the approximation guarantees for Algorithm 2 and its time complexity. The proof of Theorem 2 is in Appendix C. For $\mathcal{C}_t$ given by any algorithm of Table 2, $F(A) = \max_{\boldsymbol{\delta}\in\mathcal{C}_t^+ - \overline{\boldsymbol{\mu}}_{t-1}} \mathbf{e}_A^\intercal \boldsymbol{\delta}$, and $\varepsilon = 1$, the time complexity is bounded by $\mathcal{O}\big(m^2 n \log(mt)\big)$, and is thus efficient. Another algorithm enjoying an improved time complexity is provided in the next subsection, in the case where $\mathcal{A} = \mathcal{B}$.

Theorem 2. *Algorithm 2 outputs $S \in \mathcal{I}$ such that*

$$L(S) + 2(1+\varepsilon)F(S) \geq L(O) + F(O), \quad \forall O \in \mathcal{I}.$$

Algorithm 2 LocalSearch for maximizing $L + F$ on $\mathcal{I}$.

Input: $L, F, \mathcal{I}, m, \varepsilon > 0$.
Initialization: $S_{\text{init}} \in \arg\max_{A\in\mathcal{I}} L(A)$.
if $S_{\text{init}} = \emptyset$ **then**
  if $\exists\{x\} \in \mathcal{I}$ such that $(L+F)(\{x\}) > 0$ **then**
    $S_0 \in \arg\max_{\{x\}\in\mathcal{I},\ (L+F)(\{x\})>0} L(\{x\})$.
  else
    Output $\emptyset$
  end if
else
  $S_0 \leftarrow S_{\text{init}}$
end if
$S \leftarrow S_0$.
Repeatedly perform one of the following local improvements **while** possible:

- **Delete an element:**
  if $\exists x \in S$ such that
  $(L+F)(S\backslash\{x\}) > (L+F)(S) + \frac{\varepsilon}{m}F(S)$,
  then $S \leftarrow S\backslash\{x\}$.
  end if
- **Add an element:**
  if $\exists y \in [n]\backslash S,\ S \cup \{y\} \in \mathcal{I}$, such that
  $(L+F)(S\cup\{y\}) > (L+F)(S) + \frac{\varepsilon}{m}F(S)$,
  then $S \leftarrow S \cup \{y\}$.
  end if
- **Swap a pair of elements:**
  if $\exists(x,y) \in S \times [n]\backslash S,\ S\backslash\{x\}\cup\{y\} \in \mathcal{I}$, such that
  $(L+F)(S\backslash\{x\}\cup\{y\}) > (L+F)(S) + \frac{\varepsilon}{m}F(S)$
  then $S \leftarrow S\backslash\{x\}\cup\{y\}$ **end if**

end while
Output: S.

Its complexity is $\mathcal{O}\Big(mn\log\Big(\frac{\max_{A\in\mathcal{I}} F(A)}{F(S_0)}\Big)/\log\big(1+\frac{\varepsilon}{m}\big)\Big)$.

Theorem 2 gives a parameter κ arbitrary close to 2 in (4).[7]

4.2. Greedy Algorithm

In this section, we assume that $\mathcal{A} = \mathcal{B}$. This situation happens, for instance, under a non-negativity assumption on L, i.e., if we consider non-negative rewards X_i. We show that the standard Greedy algorithm (Algorithm 3) improves over Algorithm 2 by exploiting this extra constraint, both in terms of the running time and the approximation factor. We state the result in Theorem 3 and prove it in Appendix D.

[7]We could design a different version of Algorithm 2, based on Non-ObliviousLocalSearch (Filmus & Ward, 2012) , in order to get κ arbitrary close to $1 + 1/(e-1)$, but with a much worst time complexity. Actually, Sviridenko et al. (2013) proposed such an approach, with an approximation factor for L arbitrary close to 1, but not equal, so we would get back the undesirable term, which would require a complexity polynomial in t to control.

Algorithm 3 GREEDY for maximizing $L+F$ on $\mathcal{B}$.

Input: $L, F, \mathcal{I}, m$.
Initialization: $S \leftarrow \emptyset$.
for $i \in [k]$ **do**
  $x \in \arg\max_{x \notin S,\ S \cup \{x\} \in \mathcal{I}} (L+F)(S \cup \{x\})$.
  $S \leftarrow S \cup \{x\}$.
end for
Output: S.

Notice that another advantage is that we do not need to assume $F(A) > 0$ for $A \neq \emptyset$ here.

Theorem 3. *Algorithm 3 outputs $S \in \mathcal{B}$ such that*

$$L(S) + 2F(S) \geq L(O) + F(O), \quad \forall O \in \mathcal{B}.$$

Its complexity is $\mathcal{O}(mn)$.

Combining the results before, we get the following theorem.

Theorem 4. *With approximation techniques, the cumulative regret for the combinatorial semi-bandits is bounded as*

$$R_T \leq \mathcal{O}\left(\sqrt{n \log^2(m) T \log T}\right)$$

with per-round time complexity of order $\mathcal{O}(\log(mt)m^2 n)$ (resp., $\mathcal{O}(mn)$) for $\mathcal{A} = \mathcal{I}$ (resp., $\mathcal{A} = \mathcal{B}$).

Notice that this new bound is better by a factor $\sqrt{m}/\log m$ than the one of Kveton et al. (2014) in the case $\mathcal{A} = \mathcal{B}$.

4.3. Budgeted Matroid Semi-Bandits

In this subsection, we extend results of the two previous subsections to budgeted matroid semi-bandits. In budgeted bandits with single resource and infinite horizon (Ding et al., 2013; Xia et al., 2016a), each arm is associated with both a reward and a cost. The agent aims at maximizing the cumulative reward under a budget constraint for the cumulative costs. Xia et al. (2016b) studied budgeted bandits with multiple play, where a m-subset A of arms is selected at each round. An optimal (up to a constant term) offline algorithm chooses the same action $A^\star$ within each round, where $A^\star$ is the minimizer of the ratio "expected cost paid choosing A" over "expected reward gained choosing A". In the setting of Xia et al. (2016b), the agent observes the *partial* random cost and reward of each arm in A (i.e., semi-bandit feedback), pays the sum of partial costs of A and gains the sum of partial rewards of A. $A^\star$ can be computed efficiently, and a Xia et al. (2016b) give an algorithm based on CUCB. It minimizes the ratio where the averages are replaced by UCBs. We extend this setting to matroid constraints. We assume that total costs/rewards are non-negative linear set functions of the chosen action A. The objective is to minimize a ratio of linear set functions. As previously, two kinds of constraints can be considered for the minimization: either $\mathcal{A} = \mathcal{I}$ or $\mathcal{A} = \mathcal{B}$. Theorem 1 implies that an optimistic estimation of this ratio is of the form $\frac{L_1 - F_1}{L_2 + F_2}$, where for $i \in \{1, 2\}$, F_i are positive (except for $\emptyset$), normalized, non-decreasing, submodular; and L_i are non-negative and linear. $L_1 - F_1$ is a high-probability lower bound on the expected cost paid, and $L_2 + F_2$ is a high-probability upper bound on the expected reward gained. Notice that the numerator, $L_1 - F_1$, can be negative, which can be an incitement to take arms with a high cost/low rewards. Therefore, we consider the minimization of the surrogate $\left(\frac{L_1 - F_1}{L_2 + F_2}\right)^+$. Indeed, $(L_1 - F_1)/(L_2 + F_2)$ is a high probability lower bound on the ratio of expectation, so by monotonicity of $x \mapsto x^+$ on $\mathbb{R}$, $(L_1 - F_1)^+/(L_2 + F_2)$ is also a high-probability lower bound. We assume L_2 is normalized, but not necessarily L_1. $L_1(\emptyset)$ can be seen as an entry price. When L_1 is normalized, we assume that $\emptyset$ is not feasible.

Remark 2. *Notice, If $\mathcal{A} = \mathcal{I}$, and L_1 is normalized, then there is an optimal solution of the form $\{s\} \in \mathcal{I}$: If $L_1 - F_1$ is negative for some $S = \{s\} \subset \mathcal{I}$, then such S is a minimizer. Otherwise, by submodularity (and thus subadditivity, since we consider normalized functions), $L_1 - F_1$ is non-negative, and we have*

$$\begin{aligned} \frac{L_1(S) - F_1(S)}{L_2(S) + F_2(S)} &\geq \frac{\sum_{s \in S} L_1(\{s\}) - F_1(\{s\})}{\sum_{s \in S} L_2(\{s\}) + F_2(\{s\})} \\ &\geq \min_{s \in S} \frac{L_1(\{s\}) - F_1(\{s\})}{L_2(\{s\}) + F_2(\{s\})}. \end{aligned}$$

This minimization problem is at least as difficult as previous submodular maximization problems, taking $L_1 = 1$ and $F_1 = 0$. In order to use our approximation algorithms, we consider the *Lagrangian function* associated to the problem (see e.g., Fujishige, 2005),

$$\mathcal{L}(\lambda, S) \triangleq L_1(S) - F_1(S) - \lambda(L_2(S) + F_2(S)),$$

for $\lambda \geq 0$ and $S \subset [n]$. For a fixed $\lambda \geq 0$, $-\mathcal{L}(\lambda, \cdot)$ is a sum of a linear and a submodular function, and both Algorithms 2 and 3 can be used. However, the first step is to find λ sufficiently close to the optimal ratio

$$\lambda^\star = \min_{A \in \mathcal{A}} \left(\frac{L_1(A) - F_1(A)}{L_2(A) + F_2(A)}\right)^+.$$

Remark 3. *For some $\lambda \geq 0$,*

$$\min_{A \in \mathcal{A}} \mathcal{L}(\lambda, A) \geq 0 \Rightarrow \lambda \leq \lambda^\star,$$

$$\min_{A \in \mathcal{A}} \mathcal{L}(\lambda, A) \leq 0 \Rightarrow \begin{cases} \lambda \geq \lambda^\star, \textit{ or} \\ \min_{A \in \mathcal{A}} L_1(A) - F_1(A) \leq 0, \\ \textit{which further gives } \lambda^\star = 0. \end{cases}$$

From Remark 3, if it was possible to compute $\min_{A \in \mathcal{A}} \mathcal{L}(\lambda, A)$ exactly, then a binary search algorithm

Algorithm 4 Binary search for minimizing the ratio $(L_1 - F_1)^+/(L_2 + F_2)$.

Input: L_1, L_2, F_1, F_2, $\textsc{Algo}_\kappa$, $\eta > 0$.
$\delta \leftarrow \frac{\eta \min_{\{s\} \in \mathcal{A}} F_1(\{s\})}{L_2(B) + \kappa^2 F_2(B)}$ with $B = \textsc{Algo}_\kappa(L_2 + \kappa F_2)$.
$A \leftarrow A_0 \in \mathcal{A} \backslash \{\emptyset\}$ arbitrary.
if $\mathcal{L}_\kappa(0, A) > 0$ **then**
  $\lambda_1 \leftarrow 0, \quad \lambda_2 \leftarrow \frac{L_1(A) - F_1(A)}{L_2(A) + F_2(A)}$.
  while $\lambda_2 - \lambda_1 \geq \delta$ **do**
    $\lambda \leftarrow \frac{\lambda_1 + \lambda_2}{2}$.
    $S \leftarrow \textsc{Algo}_\kappa(-\mathcal{L}(\lambda, \cdot))$.
    if $\mathcal{L}_\kappa(\lambda, S) \geq 0$ **then**
      $\lambda_1 \leftarrow \lambda$.
    else
      $\lambda_2 \leftarrow \lambda$.
      $A \leftarrow S$.
    end if
  end while
end if
Output: A.

would find $\lambda^\star$. This dichotomy method can be extended to $\kappa-$approximation algorithms by defining the *approximation Lagrangian* as

$$\mathcal{L}_\kappa(\lambda, S) \triangleq L_1(S) - \kappa F_1(S) - \lambda(L_2(S) + \kappa F_2(S)),$$

for $\lambda \geq 0$ and $S \subset [n]$. The idea is to use the following approximation guarantee for a $\kappa-$approximation algorithms outputing S,

$$\min_{A \in \mathcal{A}} \mathcal{L}_\kappa(\lambda, A) \leq \mathcal{L}_\kappa(\lambda, S) \leq \min_{A \in \mathcal{A}} \mathcal{L}(\lambda, A).$$

Thus, for a given λ, either the l.h.s is strictly negative or the r.h.s is non-negative, depending on the sign of $\mathcal{L}_\kappa(\lambda, S)$.

Therefore, from Remark 3, a lower bound λ_1 on $\lambda^\star$, and an upper bound λ_2 on $\min_{A \in \mathcal{A}} \left(\frac{L_1(A) - \kappa F_1(A)}{L_2(A) + \kappa F_2(A)}\right)^+$ can be computed. The detailed method is given in Algorithm 4. Notice that it takes as input some $\textsc{Algo}_\kappa$, that can be either Algorithm 2 or Algorithm 3, depending on the assumption on the constraint (either $\mathcal{A} = \mathcal{I}$ or $\mathcal{A} = \mathcal{B}$). We denote the output as $\textsc{Algo}_\kappa(L + F)$, for some linear set function L and some submodular set function F, for maximizing the objective $L + F$ on $\mathcal{A}$, so that $S = \textsc{Algo}_\kappa(L + F)$ satisfies $L(S) + \kappa F(S) \geq \max_{A \in \mathcal{A}} L(A) + F(A)$, i.e., $\kappa = 2(1 + \varepsilon)$ if $\textsc{Algo}_\kappa =$ Algorithm 2, $\mathcal{A} = \mathcal{I}$ and $\kappa = 2$ if $\textsc{Algo}_\kappa =$ Algorithm 3, $\mathcal{A} = \mathcal{B}$. In Theorem 5, we state the result for the output of Algorithm 4 and prove it in Appendix E.

Theorem 5. *Algorithm 4 outputs A such that*

$$\left(\frac{L_1(A) - (\kappa + \eta) F_1(A)}{L_2(A) + \kappa F_2(A)}\right)^+ \leq \lambda^\star,$$

where $\lambda^\star$ is the minimum of $\left(\frac{L_1 - F_1}{L_2 + F_2}\right)^+$ over $\mathcal{I}$ if $\textsc{Algo}_\kappa =$ Algorithm 2, and over $\mathcal{B}$ if $\textsc{Algo}_\kappa =$ Algorithm 3. For $\mathcal{C}_t$ given by any algorithm of Table 2, $F(A) = \max_{\boldsymbol{\delta} \in \mathcal{C}_t^+ - \overline{\boldsymbol{\mu}}_{t-1}} \mathbf{e}_A^\mathsf{T} \boldsymbol{\delta}$, the complexity is of order $\log(mt/\eta)$ times the complexity of $\textsc{Algo}_\kappa$.

5. Experiments

We provide experiments for a *graphic matroid*, on a five nodes complete graph, as did Combes et al. (2015). We thus have $n = 10$, $m = 4$. We consider two experiments. In the first one we use $\mathcal{A} = \mathcal{B}$, $\mu_i = 1 + \Delta \mathbb{I}\{i \leq m\}$, for all $i \in [n]$, and in the second, $\mathcal{A} = \mathcal{I}$, where we set $\mu_i = \Delta(2\mathbb{I}\{i \leq m - 1\} - 1)$, $\forall i \in [n]$. We take $\Delta = 0.1$, with rewards drawn from independent unit variance Gaussian distributions. Figure 1 illustrates the comparison between CUCB and our implementations of ESCB (Combes et al., 2015) using Algorithm 3 (left) and 2 (right, with $\varepsilon = 0.1$), showing the behavior of the regret vs. time horizon. We observe that although we are approximating the confidence region within a factor at least 2 (and thus force the exploration), our efficient implementation outperforms CUCB. Indeed, we take advantage (gaining a factor $\sqrt{m}/\log m$) of the previously inefficient algorithm (that we made efficient) to use reward independence, which the more conservative CUCB is not able to. The latter algorithm has still a better per round-time complexity of $\mathcal{O}(n \log m)$ and may be more practical on larger instances.

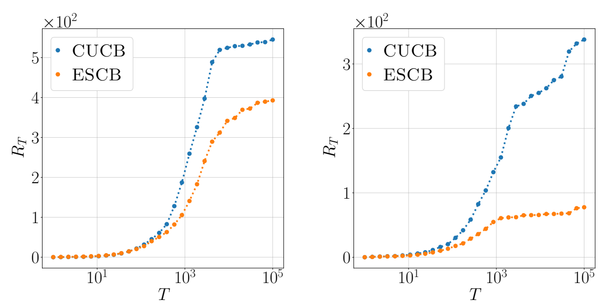

Figure 1. Cumulative regret for the minimum spanning tree setting in up to 10^5 rounds, averaged over 100 independent simulations. **Left**: for $\mathcal{A} = \mathcal{B}$. **Right**: for $\mathcal{A} = \mathcal{I}$.

6. Discussion

In this paper, we gave several approximation schemes for the confidence regions and applied them to combinatorial semi-bandits with matroid constraints and their budgeted version. We believe our approximation methods can be extended to approximation regret for non-linear objective functions (e.g., for influence maximization, Wang & Chen, 2018), if the maximization algorithm keeps the same approximation factor for the objective, either with or without the bonus.

Acknowledgments Vianney Perchet has benefited from the support of the ANR (grant n.ANR-13-JS01-0004-01), of the FMJH Program Gaspard Monge in optimization and operations research (supported in part by EDF), from the Labex LMH and from the CNRS through the PEPS program. The research presented was also supported by European CHIST-ERA project DELTA, French Ministry of Higher Education and Research, Nord-Pas-de-Calais Regional Council, Inria and Otto-von-Guericke-Universität Magdeburg associated-team north-European project Allocate, and French National Research Agency project BoB (grant n.ANR-16-CE23-0003), FMJH Program PGMO with the support to this program from Criteo.

References

Audibert, J.-Y., Bubeck, S., and Lugosi, G. Regret in online combinatorial optimization. *Mathematics of Operations Research*, 39:31–45, 2014.

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

Bach, F. Learning with Submodular Functions: A Convex Optimization Perspective. 2011.

Brualdi, R. A. Comments on bases in dependence structures. *Bulletin of the Australian Mathematical Society*, 1(2):161–167, 1969.

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

Chen, W., Wang, Y., and Yuan, Y. Combinatorial Multi-Armed Bandit: General Framework and Applications. In *International Conference on Machine Learning*, pp. 151–159, 2013.

Chen, W., Wang, Y., Yuan, Y., and Wang, Q. Combinatorial Multi-Armed Bandit and Its Extension to Probabilistically Triggered Arms. jul 2014.

Combes, R., Shahi, M. S. T. M., Proutiere, A., and Others. Combinatorial bandits revisited. In *Advances in Neural Information Processing Systems*, pp. 2116–2124, 2015.

Degenne, R. and Perchet, V. Combinatorial semi-bandit with known covariance. dec 2016.

Ding, W., Qin, T., Zhang, X.-d., and Liu, T.-y. Multi-Armed Bandit with Budget Constraint and Variable Costs. In *Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013.

Edmonds, J. Matroids and the Greedy Algorithm. *Mathematical Programming*, 1(1):127–136, 1971.

Edmonds, J. and Fulkerson, D. Transversals and matroid partition. *Journal of Research of the National Bureau of Standards Section B Mathematics and Mathematical Physics*, 69B(3):147–153, 1965.

Filmus, Y. and Ward, J. A tight combinatorial algorithm for submodular maximization subject to a matroid constraint. In *Proceedings - Annual IEEE Symposium on Foundations of Computer Science, FOCS*, pp. 659–668, 2012.

Fujishige, S. *Submodular functions and optimization*. Annals of discrete mathematics. 2005.

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

He, S., Zhang, J., and Zhang, S. Polymatroid Optimization, Submodularity, and Joint Replenishment Games. *Operations Research*, 60(1):128–137, 2012.

Jun, K.-S., Bhargava, A., Nowak, R., and Willett, R. Scalable Generalized Linear Bandits: Online Computation and Hashing. may 2017.

Koolen, W., Warmuth, M. K., and Kivinen, J. Hedging structured concepts. In *COLT*, volume 118653, pp. 93–105, 2010.

Krause, A. and Golovin, D. Submodular function maximization. In *Tractability*, volume 9781107025, pp. 71–104. 2011.

Kveton, B., Wen, Z., Ashkan, A., Eydgahi, H., and Eriksson, B. Matroid bandits: Fast combinatorial optimization with learning. In *Uncertainty in Artificial Intelligence*, 2014.

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

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

Lee, J., Mirrokni, V. S., Nagarajan, V., and Sviridenko, M. Maximizing Nonmonotone Submodular Functions under Matroid or Knapsack Constraints. *SIAM Journal on Discrete Mathematics*, 23(4):2053–2078, 2010.

Li, L., Chu, W., Langford, J., and Schapire, R. E. A contextual-bandit approach to personalized news article recommendation. *International World Wide Web Conference*, 2010.

Mohri, M. and Munoz, A. Optimal regret minimization in posted-price auctions with strategic buyers. In *Advances in Neural Information Processing Systems*, pp. 1871–1879, 2014.

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

Neu, G. and Bartók, G. An efficient algorithm for learning with semi-bandit feedback. In *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 8139 LNAI, pp. 234–248, 2013.

Oliveira, C. A. and Pardalos, P. M. A survey of combinatorial optimization problems in multicast routing, 2005.

Perfect, H. Applications of Menger's graph theorem. *Journal of Mathematical Analysis and Applications*, 22(1): 96–111, 1968.

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

Russo, D. and Roy, B. V. An information-theoretic analysis of thompson sampling. *Journal of Machine Learning Research*, 17(68):1–30, 2016.

Russo, D. and Van Roy, B. Learning to Optimize Via Posterior Sampling. jan 2013.

Sakaue, S., Ishihata, M., and Minato, S.-i. Efficient Bandit Combinatorial Optimization Algorithm with Zero-suppressed Binary Decision Diagrams. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, volume 84, pp. 585–594, 2018.

Schrijver, A. *Combinatorial Optimization: Polyhedra and Efficiency*. 2008.

Stobbe, P. and Krause, A. Efficient Minimization of Decomposable Submodular Functions. oct 2010.

Sviridenko, M., Vondrák, J., and Ward, J. Optimal approximation for submodular and supermodular optimization with bounded curvature. nov 2013.

Talebi, M. S. and Proutiere, A. An Optimal Algorithm for Stochastic Matroid Bandit Optimization. In *The 2016 International Conference on Autonomous Agents & Multiagent Systems*, pp. 548–556, 2016.

Talebi, M. S., Zou, Z., Combes, R., Proutiere, A., and Johansson, M. Stochastic Online Shortest Path Routing: The Value of Feedback. sep 2013.

Tran-Thanh, L., Stein, S., Rogers, A., and Jennings, N. R. Efficient crowdsourcing of unknown experts using bounded multi-armed bandits. *Artificial Intelligence*, 214: 89–111, 2014.

Wang, S. and Chen, W. Thompson Sampling for Combinatorial Semi-Bandits. mar 2018.

Whitney, H. On the abstract properties of linear dependence. *American Journal of Mathematics*, 57(3):509–533, 1935.

Xia, Y., Ding, W., Zhang, X.-D., Yu, N., and Qin, T. Budgeted bandit problems with continuous random costs. In *Asian Conference on Machine Learning*, 2016a.

Xia, Y., Qin, T., Ma, W., Yu, N., and Liu, T.-Y. Budgeted multi-armed bandits with multiple plays. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pp. 2210–2216. AAAI Press, 2016b.

A. Proof of Proposition 1

Proof. It suffices to maximize on the coordinates of $\boldsymbol{\delta}$ belonging to A (the others being zero). For all $i \in A$, we let

$$\eta_i^\star \triangleq \left(1 - \lambda^\star g'_{i,t}(r - \overline{\mu}_{i,t-1})\right)\mathbb{I}\left\{\delta_i^\star = r - \overline{\mu}_{i,t-1}\right\}$$
$$\gamma_i^\star \triangleq \left(\lambda^\star g'_{i,t}(0) - 1\right)\mathbb{I}\{\delta_i^\star = 0\} = -\mathbb{I}\{\delta_i^\star = 0\}.$$

For all $i \in A$, the function f_i is continuous, non-increasing on $\mathbb{R}_+$, hence so is $\lambda \mapsto \mathbf{e}_A^\intercal \left(g_{i,t}(f_i(\lambda))\right)_i$. If $\mathbf{e}_A^\intercal \left(g_{i,t}(f_i(\lambda^\star))\right)_i < 1$, then necessarily $\lambda^\star = 0$. Thus, the following KKT conditions are satisfied:

$$\begin{aligned}
\lambda^\star \left(\sum_{i \in A} g_{i,t}(\delta_i^\star) - 1\right) &= 0, \text{ and} \\
\forall i \in A,\ \lambda^\star g'_{i,t}(\delta_i^\star) + \eta_i^\star - \gamma_i^\star &= 1, \\
\eta_i^\star(\delta_i^\star - r + \overline{\mu}_{i,t-1}) &= 0, \\
-\gamma_i^\star \delta_i^\star &= 0,
\end{aligned}$$

which concludes the proof by the convexity of the constraints and the objective function. □

B. Proof of Theorem 1

Proof. Let $t \in \mathbb{N}^\star$. We consider here the restriction of $g_{i,t}$ to $[0, r - \overline{\mu}_{i,t-1}]$, that we still denote as $g_{i,t}$. Notice that for all $i \in [n]$, $g_{i,t}$ is either 0 or a bijection on $[0, r - \overline{\mu}_{i,t-1}]$ by assumption. For $p = \infty$, we have that

$$\max_{\boldsymbol{\delta} \in \mathcal{C}_t^+ - \overline{\boldsymbol{\mu}}_{t-1}} \mathbf{e}_A^\intercal \boldsymbol{\delta} = \mathbf{e}_A^\intercal \left(\min\left\{g_{i,t}^{-1}(1), r - \overline{\mu}_{i,t-1}\right\}\mathbb{I}\{N_{i,t-1} \geq 1\} + r\mathbb{I}\{N_{i,t-1} = 0\}\right)_i$$

is a linear set function of A. Assume now that $p = 1$. To show the submodularity of $A \mapsto \max_{\boldsymbol{\delta} \in \mathcal{C}_t^+ - \overline{\boldsymbol{\mu}}_{t-1}} \mathbf{e}_A^\intercal \boldsymbol{\delta}$ in this case, we will use the notion of *polymatroid*.

Definition 3 (Polymatroid)**.** *A polymatroid is a polytope of the forme $\left\{\boldsymbol{\delta}' \in \mathbb{R}_+^n, \mathbf{e}_A^\intercal \boldsymbol{\delta}' \leq F(A),\ \forall A \subset [n]\right\}$, where F is a non-decreasing submodular function.*

Fact 1 (Theorem 3 of He et al., 2012)**.** *Let P be a polymatroid, and let $h_1, \dots, h_n$ be concave functions. Then $A \mapsto \max_{\boldsymbol{\delta}' \in P} \mathbf{e}_A^\intercal \left(h_i(\delta')\right)_i$ is submodular.*

Notice that $g_{i,t}^{-1}(\{0\}) = [0, r - \overline{\mu}_{i,t-1}]$ when $N_{i,t-1} = 0$, and that $g_{i,t}^{-1}(\cdot)$ is a strictly increasing concave function on $[0, g_{i,t}(r - \overline{\mu}_{i,t-1})]$, as the inverse function of a strictly increasing convex function when $N_{i,t-1} \geq 1$. So we can rewrite $\mathcal{C}_t^+ - \overline{\boldsymbol{\mu}}_{t-1}$ as an union of product sets:

$$\mathcal{C}_t^+ - \overline{\boldsymbol{\mu}}_{t-1} = \left\{\boldsymbol{\delta} \in \prod_{i \in [n]} [0, r - \overline{\mu}_{i,t-1}],\ \sum_{i \in [n]} g_{i,t}(\delta_i) \leq 1\right\} = \bigcup_{\substack{\boldsymbol{\delta}' \in \prod_{i \in [n]} [0, g_{i,t}(r - \overline{\mu}_{i,t-1})], \\ \sum_{i \in [n]} \delta'_i \leq 1}} \prod_{i \in [n]} g_{i,t}^{-1}(\{\delta'_i\}).$$

We can thus rewrite our function as

$$\max_{\boldsymbol{\delta} \in \mathcal{C}_t^+ - \overline{\boldsymbol{\mu}}_{t-1}} \mathbf{e}_A^\intercal \boldsymbol{\delta} = \max_{\substack{\boldsymbol{\delta}' \in \prod_{i \in [n]} [0, g_{i,t}(r - \overline{\mu}_{i,t-1})], \\ \sum_{i \in [n]} \delta'_i \leq 1}} \mathbf{e}_A^\intercal \left(g_{i,t}^{-1}(\delta'_i)\right)_i,$$

with the convention $g_{i,t}^{-1}(0) = r - \overline{\mu}_{i,t-1}$ when $N_{i,t-1} = 0$.

The constraints' set $\left\{\boldsymbol{\delta}' \in \prod_{i \in [n]} [0, g_{i,t}(r - \overline{\mu}_{i,t-1})],\ \sum_{i \in [n]} \delta'_i \leq 1\right\}$ is equal to the intersection between $\prod_{i \in [n]} [0, g_{i,t}(r - \overline{\mu}_{i,t-1})]$ and the polymatroid $\left\{\boldsymbol{\delta}' \in \mathbb{R}_+^n, \mathbf{e}_A^\intercal \boldsymbol{\delta}' \leq \mathbb{I}\{A \neq \emptyset\},\ \forall A \subset [n]\right\}$. This intersection is itself equal to the polymatroid $\left\{\boldsymbol{\delta}' \in \mathbb{R}_+^n, \mathbf{e}_A^\intercal \boldsymbol{\delta}' \leq \min_{B \subset A}\left\{\mathbb{I}\{B \neq A\} + \mathbf{e}_B^\intercal \left(g_{i,t}(r - \overline{\mu}_{i,t-1})\right)_i\right\},\ \forall A \subset [n]\right\}$.

Thus, $\max_{\boldsymbol{\delta} \in \mathcal{C}_t^+ - \overline{\boldsymbol{\mu}}_{t-1}} \mathbf{e}_A^\intercal \boldsymbol{\delta}$ is the optimal objective value on a polymatroid of a separable concave function, as a function of the index set A. Now, using Fact 1, it is submodular. □

C. Proof of Theorem 2

Before proving Theorem 2, we state some well known results about submodular optimization on a matroid.

Proposition 2. *Let $A, B \subset [n]$. If F is submodular, then*

$$\sum_{b\in B\backslash A}(F(B)-F(B\backslash\{b\}))\leq F(B)-F(A\cap B),\qquad \sum_{a\in A\backslash B}(F(B\cup\{a\})-F(B))\geq F(A\cup B)-F(B).$$

Proof. Let $(b_1,\dots,b_{|B\backslash A|})$ be an ordering of $B\backslash A$. Then, by submodularity of F,

$$\sum_{i=1}^{|B\backslash A|}(F(B)-F(B\backslash\{b_i\}))\leq\sum_{i=1}^{|B\backslash A|}(F(B\backslash\{b_1,\dots,b_{i-1}\})-F(B\backslash\{b_1,\dots,b_i\}))=F(B)-F(A\cap B).$$

In the same way, let $(a_1,\dots,a_{|A\backslash B|})$ be an ordering of $A\backslash B$. Then, by submodularity of F,

$$\sum_{i=1}^{|A\backslash B|}(F(B\cup\{a_i\})-F(B))\geq\sum_{i=1}^{|A\backslash B|}(F(B\cup\{a_1,\dots,a_i\})-F(B\cup\{a_1,\dots,a_{i-1}\}))=F(A\cup B)-F(B).$$

□

Fact 2 (Theorem 1 of Lee et al., 2010)**.** *Let $A, B\in\mathcal{A}$. Then, there exists a mapping $\alpha: B\backslash A\rightarrow A\backslash B\cup\{\emptyset\}$ such that*

- $\forall b\in B\backslash A,\ A\backslash\{\alpha(b)\}\cup b\in\mathcal{A}$
- $\forall a\in A\backslash B,\ \left|\alpha^{-1}(a)\right|\leq 1.$

Proposition 3. *Let $A, B\in\mathcal{A}$. Let F be a submodular function and $\alpha: B\backslash A\rightarrow A\backslash B\cup\{\emptyset\}$ be the mapping given in Fact 2. Then,*

$$\sum_{b\in B\backslash A}(F(A)-F(A\backslash\{\alpha(b)\}\cup\{b\}))+\sum_{a\in A\backslash B,\ \alpha^{-1}(a)=\emptyset}(F(A)-F(A\backslash\{a\}))\leq 2F(A)-F(A\cup B)-F(A\cap B).$$

Proof. We decompose $\sum_{b\in B\backslash A}(F(A)-F(A\backslash\{\alpha(b)\}\cup\{b\}))$ into sum of two terms,

$$\sum_{b\in B\backslash A}(F(A)-F(A\backslash\{\alpha(b)\}))+\sum_{b\in B\backslash A}(F(A\backslash\{\alpha(b)\})-F(A\backslash\{\alpha(b)\}\cup\{b\})).$$

Remark that the first part is equal to

$$\sum_{a\in\alpha(B\backslash A)}(F(A)-F(A\backslash\{a\}))=\sum_{a\in A\backslash B,\ \alpha^{-1}(a)\neq\emptyset}(F(A)-F(A\backslash\{a\})).$$

Thus, together with $\sum_{a\in A\backslash B,\ \alpha^{-1}(a)=\emptyset}(F(A)-F(A\backslash\{a\}))$, we get that

$$\sum_{b\in B\backslash A}(F(A)-F(A\backslash\{\alpha(b)\}\cup\{b\}))+\sum_{a\in A\backslash B,\ \alpha^{-1}(a)=\emptyset}(F(A)-F(A\backslash\{a\}))$$

is equal to

$$\sum_{a\in A\backslash B}(F(A)-F(A\backslash\{a\}))+\sum_{b\in B\backslash A}(F(A\backslash\{\alpha(b)\})-F(A\backslash\{\alpha(b)\}\cup\{b\})).$$

Finally, we upper bound the first term by $F(A)-F(A\cap B)$ using first inequality of Lemma 2, and the second term by $F(A)-F(A\cup B)$ using first, the submodularity of F to remove $\alpha(b)$ in the summands, and then the second inequality of Lemma 2. □

Proof of Theorem 2. The proof is divided into two parts:

Approximation guarantee If Algorithm 2 outputs $\emptyset$ before entering in the while loop, then by submodularity, for any $S \in \mathcal{I}$,

$$(L+F)(S) \leq \sum_{x \in S} (L+F)(\{x\}) \leq 0.$$

Thus, $\emptyset$ is a maximizer of $L+F$.

Otherwise, the output S of Algorithm 2 satisfies the local optimality of the while loop. We apply Proposition 3 with $A = S$ and $B = O$ for L and F separately,

$$\sum_{b \in O \setminus S} (L(S) - L(S \setminus \{\alpha(b)\} \cup \{b\})) + \sum_{a \in S \setminus O,\ \alpha^{-1}(a) = \emptyset} (L(S) - L(S \setminus \{a\})) \leq 2L(S) - L(S \cup O) - L(S \cap O),$$

$$\sum_{b \in O \setminus S} (F(S) - F(S \setminus \{\alpha(b)\} \cup \{b\})) + \sum_{a \in S \setminus O,\ \alpha^{-1}(a) = \emptyset} (F(S) - F(S \setminus \{a\})) \leq 2F(S) - F(S \cup O) - F(S \cap O).$$

Then, we sum these two inequalities,

$$\begin{aligned}\sum_{b \in O \setminus S} ((L+F)(S) - (L+F)(S \setminus \{\alpha(b)\} \cup \{b\})) + \sum_{a \in S \setminus O,\ \alpha^{-1}(a) = \emptyset} &((L+F)(S) - (L+F)(S \setminus \{a\})) \\ &\leq 2(L+F)(S) - (L+F)(S \cup O) - (L+F)(S \cap O) \\ &= 2F(S) - F(S \cup O) - F(S \cap O) + L(S) - L(O),\end{aligned}$$

where the last equality uses linearity of L. Since F is increasing and non-negative, $F(S \cup O) + F(S \cap O) \geq F(O)$, and we get

$$\begin{aligned}\sum_{b \in O \setminus S} ((L+F)(S) - (L+F)(S \setminus \{\alpha(b)\} \cup \{b\})) + \sum_{a \in S \setminus O,\ \alpha^{-1}(a) = \emptyset} &((L+F)(S) - (L+F)(S \setminus \{a\})) \\ &\leq 2F(S) - F(O) + L(S) - L(O).\end{aligned}$$

From the local optimality of S, the left hand term in this inequality is lower bounded by

$$\sum_{b \in O \setminus S} \frac{-\varepsilon}{m} F(S) + \sum_{a \in S \setminus O,\ \alpha^{-1}(a) = \emptyset} \frac{-\varepsilon}{m} F(S) \geq -2\varepsilon F(S).$$

The last statement finishes the proof for the approximation inequality.

Time complexity Computing S_0 has a negligible complexity compared to the while loop. The following lemma gives a characterization of S_0.

Lemma 2. $S_0 \in \arg\max\{L(A),\ A \in \mathcal{I},\ (F+L)(A) > 0\}$.

Proof. From Algorithm 2, if $S_{\text{init}} \neq \emptyset$, then $S_0 = S_{\text{init}}$ and $L(S_0) = \max_{A \in \mathcal{I}} L(A) \geq 0$. Thus, $F(S_0) > 0$ by assumption on F, giving $(F+L)(S_0) > 0$, which ends the proof. If $S_{\text{init}} = \emptyset$, then $L(S_0) = \max\{L(\{x\}),\ \{x\} \in \mathcal{I},\ (L+F)(\{x\}) > 0\}$. Let $A \in \arg\max\{L(A),\ A \in \mathcal{I},\ (F+L)(A) > 0\}$. A is clearly non-empty, and by submodularity of $F+L$, there exists $x \in A$ such that $(F+L)(\{x\}) > 0$. L is non-increasing from $S_{\text{init}} = \emptyset$, so we get $L(\{x\}) \geq L(A)$, which means there is a singleton $\{x\}$ in $\arg\max\{L(A),\ A \in \mathcal{I},\ (F+L)(A) > 0\}$, so $S_0 \in \arg\max\{L(A),\ A \in \mathcal{I},\ (F+L)(A) > 0\}$, which finishes the proof. ☐

From this lemma, necessarily $L(S_0) \geq L(S_\ell)$ for every iterations $\ell \geq 1$, since the sequence $(L(S_\ell) + F(S_\ell))_\ell$ is increasing, and thus $(F+L)(S_\ell) > 0$, $\forall \ell \geq 1$. At each iteration $\ell \geq 1$, Algorithm 2 constructs S_ℓ such that

$$F(S_\ell) > \Big(1 + \frac{\varepsilon}{m}\Big) F(S_{\ell-1}) + L(S_{\ell-1}) - L(S_\ell).$$

Thus, we must have

$$
\begin{aligned}
F(S_\ell) - \Big(1+\frac{\varepsilon}{m}\Big)^{\ell} F(S_0) &\geq \sum_{j=1}^{\ell} \Big(1+\frac{\varepsilon}{m}\Big)^{\ell-j} (L(S_{j-1}) - L(S_j)) \\
&= L(S_0)\Big(1+\frac{\varepsilon}{m}\Big)^{\ell-1} - \frac{\varepsilon}{m}\sum_{j=1}^{\ell-1} L(S_j)\Big(1+\frac{\varepsilon}{m}\Big)^{\ell-j-1} - L(S_\ell) \\
&\geq L(S_0)\Big(1+\frac{\varepsilon}{m}\Big)^{\ell-1} - \frac{\varepsilon}{m}\sum_{j=1}^{\ell-1} L(S_0)\Big(1+\frac{\varepsilon}{m}\Big)^{\ell-j-1} - L(S_0) = 0,
\end{aligned}
$$

where the last inequality uses $L(S_0) \geq L(S_\ell),\ \forall \ell \geq 1$. This gives the following upper bound on the number of iteration ℓ:

$$
\ell \leq \frac{\log\Big(\frac{F(S_\ell)}{F(S_0)}\Big)}{\log\big(1+\frac{\varepsilon}{m}\big)} \leq \frac{\log\Big(\frac{\max_{A\in\mathcal{A}} F(A)}{F(S_0)}\Big)}{\log\big(1+\frac{\varepsilon}{m}\big)}.
$$

Finally, the result follows remarking that time complexity per iteration is $\mathcal{O}(mn)$. ☐

D. Proof of Theorem 3

As we did in the previous section, before starting the proof of Theorem 3, we state some useful results.

Fact 3 (Brualdi's lemma)**.** *Let $A, B \in \mathcal{B}$. Then, there exists a bijection $\beta : A \to B$ such that*

$$
\forall a \in A,\ A\backslash\{a\} \cup \{\beta(a)\} \in \mathcal{B}.
$$

Furthermore, β is the identity on $A \cap B$.

Proof. A proof is given by Brualdi (1969) and is also proved by Schrijver (2008), as Corollary 39.12a. ☐

Proposition 4. *Let $A, B \in \mathcal{B}$. Let F be a submodular function and $\beta : A \to B$ be the mapping given in Fact 3. Let $a_1, \dots, a_k$ be elements of A, and $A_i = \{a_1, \dots, a_i\}$. Then,*

$$
\sum_{i\in[k]} (F(A_i) - F(A_{i-1} \cup \{\beta(a_i)\})) \leq 2F(A) - F(A \cup B) - F(\emptyset).
$$

Proof. We can split $\sum_{i\in[k]}(F(A_{i-1} \cup \{a_i\}) - F(A_{i-1} \cup \{\beta(a_i)\}))$ into two terms,

$$
\sum_{i=1}^{k}(F(A_{i-1} \cup \{a_i\}) - F(A_{i-1})) + \sum_{i=1}^{k}(F(A_{i-1}) - F(A_{i-1} \cup \{\beta(a_i)\})).
$$

The first term is equal to $F(A_k) - F(\emptyset)$. Using submodularity of F, the second term is upper bounded by

$$
\sum_{i=1}^{k}(F(A_m) - F(A_m \cup \{\beta(a_i)\})),
$$

which is upper bounded by $F(A_k) - F(A_k \cup B)$ thanks to Proposition 2 and its second inequality. ☐

Proof of Theorem 3. The time complexity proof is trivial. Let $S_i \triangleq \{s_1, \dots, s_i\}$ be the set maintained in Algorithm 3 after i iterations. Instatianting Proposition 4 with $A_i = S_i$ and $B = O$, we have

$$
\sum_{i\in[k]} (F(S_i) - F(S_{i-1} \cup \{\beta(s_i)\})) \leq 2F(S) - F(S \cup O) - F(\emptyset). \tag{5}
$$

Furthermore, we also have, by linearity of L, and bijectivity of β,

$$\sum_{i\in[k]}(L(S_i)-L(S_{i-1}\cup\{\beta(s_i)\}))=\sum_{i\in[k]}(L(\{s_i\})-L(\{\beta(s_i)\})))=L(S)-L(O). \tag{6}$$

Thus, we can sum up (5) and (6) to get

$$\begin{aligned}\sum_{i\in[k]}((L+F)(S_i)-(L+F)(S_{i-1}\cup\{\beta(s_i)\}))&\le 2F(S)-F(S\cup O)-F(\emptyset)+L(S)-L(O)\\&\le 2F(S)-F(O)+L(S)-L(O),\end{aligned}$$

where the last inequality uses the fact that F is increasing and $F(\emptyset)=0$. We finish the proof remarking that by definition of Algorithm 3, $(L+F)(S_i)-(L+F)(S_{i-1}\cup\{\beta(s_i)\})\ge 0$. □

E. Proof of Theorem 5

Proof. Let A be the output of Algorithm 4 and let

$$\mathcal{L}_{\kappa_1,\kappa_2}(\lambda,S)\triangleq L_1(S)-\kappa_1F_1(S)-\lambda(L_2(S)+\kappa_2F_2(S)).$$

Recall that $\mathcal{L}_\kappa=\mathcal{L}_{\kappa,\kappa}$. Algorithm 4 satisfies either $\mathcal{L}_\kappa(0,A_0)\le 0$ — in which case Theorem 5 is trivial since

$$\left(\frac{L_1(A)-(\kappa+\eta)F_1(A)}{L_2(A)+\kappa F_2(A)}\right)^+=\lambda^\star=0$$

— or $\mathcal{L}_\kappa(0,A_0)>0$, in which case we have

$$0>\mathcal{L}_\kappa(\lambda_2,A)\ge\mathcal{L}_{\kappa+\eta,\kappa}(\lambda_2-\delta,A)\ge\mathcal{L}_{\kappa+\eta,\kappa}(\lambda_1,A)\ge\mathcal{L}_{\kappa+\eta,\kappa}(\lambda^\star,A). \tag{7}$$

The first inequality is comes from the update of λ_2: Notice that before the while loop, we have

$$\lambda_2=\frac{L_1(A_0)-F_1(A_0)}{L_2(A_0)+F_2(A_0)}>\frac{L_1(A_0)-\kappa F_1(A_0)}{L_2(A_0)+\kappa F_2(A_0)}>0,$$

since $F_2(A_0)>0$, so $0>\mathcal{L}_\kappa(\lambda_2,A_0)$ multiplying by $L_2(A_0)+F_2(A_0)$ on both sides. Notice that in particular, this inequality gives that $A\neq\emptyset$.

The second inequality follows from

$$\delta=\frac{\eta\min_{\{s\}\in\mathcal{A}}F_1(\{s\})}{L_2(B)+\kappa^2F_2(B)}\le\frac{\eta F_1(A)}{L_2(A)+\kappa F_2(A)}\quad\text{since } A\neq\emptyset \text{ and } L_2(B)+\kappa^2F_2(B)\ge L_2(A)+\kappa F_2(A).$$

Thus, multiplying by $L_2(A)+\kappa F_2(A)>0$, and adding $L_1(A)-\kappa F_1(A)-\lambda_2(L_2(A)+\kappa F_2(A))$ gives

$$L_1(A)-(\kappa+\eta)F_1(A)-(\lambda_2-\delta)(L_2(A)+\kappa F_2(A))\le L_1(A)-\kappa F_1(A)-\lambda_2(L_2(A)+\kappa F_2(A)),$$

i.e., $\mathcal{L}_{\kappa+\eta,\kappa}(\lambda_2-\delta,A)\le\mathcal{L}_\kappa(\lambda_2,A)$.

The third inequality uses $\lambda_2-\lambda_1\le\delta$, and the last inequality uses $\lambda_1\le\lambda^\star$. Indeed, since $\mathcal{L}_\kappa(\lambda_1,S)\ge 0$, the approximation relation given by $\textsc{Algo}_\kappa$,

$$\mathcal{L}_\kappa(\lambda_1,S)\le\mathcal{L}(\lambda_1,O),$$

where O is the minimizer of $\left(\frac{L_1-F_1}{L_2+F_2}\right)^+$ (for the constraints considered by $\textsc{Algo}_\kappa$), gives $0\le\mathcal{L}(\lambda_1,O)$. Thus,

$$\mathcal{L}^+(\lambda_1,O)\triangleq(L_1(O)-F_1(O))^+-\lambda_1(L_2(O)+F_2(O))\ge\mathcal{L}(\lambda_1,O)\ge 0.$$

Finally, since $L_2(O)+F_2(O)>0$ ($O\neq\emptyset$), we have $\lambda_1\le\lambda^\star$.

In (7), since $A\neq\emptyset$, we have $\frac{L_1(A)-(\kappa+\eta)F_1(A)}{L_2(A)+\kappa F_2(A)}\le\lambda^\star$ and therefore, $\left(\frac{L_1(A)-(\kappa+\eta)F_1(A)}{L_2(A)+\kappa F_2(A)}\right)^+\le\lambda^\star$.

The time complexity for the binary search is $\mathcal{O}(\log(1/\delta))\le\mathcal{O}(\log(mt/\eta))$ for $\mathcal{C}_t$ given by any algorithm of Table 2, and $F(A)=\max_{\boldsymbol{\delta}\in\mathcal{C}_t^+-\overline{\boldsymbol{\mu}}_{t-1}}\mathbf{e}_A^\intercal\boldsymbol{\delta}$. □