An adaptive stochastic optimization algorithm for resource allocation

Xavier Fontaine, Shie Mannor, Vianney Perchet

An adaptive stochastic optimization algorithm for resource allocation

Xavier Fontaine FONTAINE@CMLA.ENS-CACHAN.FR
CMLA, ENS Cachan
CNRS, Université Paris-Saclay

Shie Mannor SHIE@EE.TECHNION.AC.IL
Faculty of Electrical Engineering
Technion, Israel Institute of Technology

Vianney Perchet VIANNEY.PERCHET@NORMALESUP.ORG
CREST, ENSAE
& Criteo AI Lab, Paris

Editors: Aryeh Kontorovich and Gergely Neu

Abstract

We consider the classical problem of sequential resource allocation where a decision maker must repeatedly divide a budget between several resources, each with diminishing returns. This can be recast as a specific stochastic optimization problem where the objective is to maximize the cumulative reward, or equivalently to minimize the regret. We construct an algorithm that is *adaptive* to the complexity of the problem, expressed in term of the regularity of the returns of the resources, measured by the exponent in the Łojasiewicz inequality (or by their universal concavity parameter). Our parameter-independent algorithm recovers the optimal rates for strongly-concave functions and the classical fast rates of multi-armed bandit (for linear reward functions). Moreover, the algorithm improves existing results on stochastic optimization in this regret minimization setting for intermediate cases.

Keywords: Stochastic optimization, online learning, adaptive algorithms, resource allocation

1. Introduction

In the classical resource allocation problem, a decision maker has a fixed amount of budget (money, energy, work, etc.) to divide between several resources. Each of these resources is assumed to produce a positive return for any amount of budget allocated to them, and zero return if no budget is allocated to them (Samuelson and Nordhaus, 2005). The resource allocation problem is an age-old problem that has been theoretically investigated by Koopman (1953) and that has attracted much attention afterwards (Salehi et al., 2016; Devanur et al., 2019) due to its numerous applications (*e.g.,* production planning or portfolio selection) described for example by Gross (1956) and Katoh and Ibaraki (1998). Other applications include cases of computer scheduling, where concurrent processes compete for common and shared resources. This is the exact same problem encountered in load distribution or in project management where several tasks have to be done and a fixed amount of money/time/workers has to be distributed between those tasks. Flexible Manufacturing Systems (FMS) are also an example of application domain of our problem (Colom, 2003) and motivate our work. Resource allocation problems arise also in the domain of wireless communications systems, for example in the new 5G networks, due to the exponential growth of wireless data (Zhang et al.,

2018). Finally utility maximization in economics is also an important application of the resource allocation problem, which explains that this problem has been particularly studied in economics, where classical assumptions have been made for centuries (Smith, 1776). One of them is the *diminishing returns assumption* that states that "adding more of one factor of production, while holding all others constant, will at some point yield lower incremental per-unit returns"[1]. This natural assumption means that the reward or utility per invested unit decreases, and can be linked to submodular optimization (Korula et al., 2018).

In this paper we consider the online resource allocation problem with diminishing returns. A decision maker has to partition, at each stage, \$1 between K resources. Each resource has an unknown reward function which is assumed to be concave and increasing. As the problem is repeated in time, the decision maker can gather information about the reward functions and sequentially learn the optimal allocation. We assume that the reward itself is not observed precisely, but rather a noisy version of the gradient is observed. As usually in sequential learning – or bandit – problems (Bubeck and Cesa-Bianchi, 2012), the natural objective is to maximize the cumulative reward, or equivalently, to minimize the difference between the obtained allocation, namely the regret.

This problem is a generalization of linear resource allocation problems, widely studied in the last decade (Lattimore et al., 2015; Dagan and Crammer, 2018), where the reward functions are assumed to be linear, instead of being concave. Those approaches borrowed ideas from linear bandits (Dani et al., 2008; Abbasi-Yadkori et al., 2011). Several UCB-style algorithms with nearly optimal regret analysis have been proposed for the linear case. More general algorithms were also developed to optimize an unknown convex function with bandit feedback (Agarwal et al., 2011; Agrawal and Devanur, 2014, 2015; Berthet and Perchet, 2017) to get a generic $\widetilde{\mathcal{O}}(\sqrt{T})$[2] regret bound which is actually unavoidable with bandit feedback (Shamir, 2013). We consider instead that the decision maker has a noisy gradient feedback, so that the regularity of the reward mappings can be leveraged to recover faster rates (than $\sqrt{T}$) of convergence when possible.

There are several recent works dealing with (adaptive) algorithms for first order stochastic convex optimization. On the contrary to classical gradient-based methods, these algorithms are agnostic and adaptive to some complexity parameters of the problem, such as the smoothness or strong convexity parameters. For example, Iouditski and Nesterov (2014) proposed an adaptive algorithm to optimize uniformly convex functions and Ramdas and Singh (2013b) generalized it with an epoch-based Gradient Descent algorithm using active learning techniques, also to minimize uniformly convex functions. Both obtain optimal bounds in $\widetilde{\mathcal{O}}\left(T^{-\rho/(2\rho-2)}\right)$ for the function-error $\|f(x_t) - f^*\|$ where f is supposed to be ρ-uniformly convex (see Subsection 2.2 for a reminder on this regularity concept). However those algorithms would only achieve a $\sqrt{T}$ regret (or even a linear regret) because they rely on a structure of phases of unnecessary lengths. So in that setting, regret minimization appears to be much more challenging than function-error minimization. To be precise, we actually consider an even weaker concept of regularity than uniform convexity: the Łojasiewicz inequality (Bierstone and Milman, 1988; Bolte et al., 2010). Our objective is to devise an algorithm that can leverage this assumption, without the prior knowledge of the Łojasiewicz exponent, *i.e.,* to construct an adaptive algorithm unlike precedent approaches (Karimi et al., 2016).

High-level description of the algorithms and organization of the paper. The algorithm we are going to introduce is based on the concept of dichotomy, or binary search, which has already

1. See https://en.wikipedia.org/wiki/Diminishing_returns
2. The $\widetilde{\mathcal{O}}(\cdot)$ notation is used to hide poly-logarithmic factors.

been slightly investigated in stochastic optimization (Burnashev and Zigangirov, 1974; Castro and Nowak, 2008; Ramdas and Singh, 2013a). The specific case of $K = 2$ resources is studied in Section 3. The algorithm proposed is quite simple: it queries a point repeatedly, until it learns the sign of the gradient of the reward function, or at least with arbitrarily high probability. Then it proceeds to the next step of a standard binary search.

We will then consider, in Section 4, the case of $K \geq 3$ resources by defining a binary tree of the K resources and handling each decision using the $K = 2$ algorithm as a black-box. Our main result can be stated as follows: if the base reward mappings of the resources are β-Łojasiewicz functions, then our algorithm has a $\widetilde{\mathcal{O}}(T^{-\beta/2})$ regret bound if $\beta \leq 2$ and $\widetilde{\mathcal{O}}(T^{-1})$ otherwise. We notice that for $\beta \leq 2$ we recover existing bounds (but for the more demanding regret instead of function-error minimization) (Iouditski and Nesterov, 2014; Ramdas and Singh, 2013b) since a ρ-uniformly convex function can be proven to be β-Łojasiewicz with $\beta = \rho/(\rho - 1)$. We complement our results with a lower bound that indicates the tightness of these bounds. Finally we corroborate our theoretical findings with some experimental results, postponed to Appendix F.

Our main contributions are the design of an efficient algorithm to solve the resource allocation problem with concave reward functions. We show that our algorithm is adaptive to the unknown complexity parameters of the reward functions. Moreover we propose a unified analysis of this algorithm for a large class of functions. It is interesting to notice that our algorithm can be seen as a first-order convex minimization algorithm for separable loss functions. The setting of separable loss functions is still common in practice, though not completely general. Furthermore we prove that our algorithm outperforms other convex minimization algorithms for a broad class of functions. Finally we exhibit links with bandit optimization and we recover classical bandit bounds within our framework, highlighting the connection between bandits theory and convex optimization.

First, let us introduce in Section 2 the following general model and the different regularity assumptions mentioned above.

2. Model and Assumptions

2.1. Problem Setting

Assume a decision maker has access to $K \in \mathbb{N}^*$ different resources. We assume naturally that the number of resources K is not too large (or infinite). At each time step $t \in \mathbb{N}^*$, the agent has to split a total budget of weight 1 and to allocate $x_k^{(t)}$ to each resource $k \in [K]$ which generates the reward $f_k(x_k(t))$. Overall, at this stage, the reward of the decision maker is then

$$F(x^{(t)}) = \sum_{k \in [K]} f_k(x_k^{(t)}) \quad \text{with } x^{(t)} = (x_1^{(t)}, \dots, x_K^{(t)}) \in \Delta^K,$$

where the simplex $\Delta^K = \left\{(p_1, \dots, p_K) \in \mathbb{R}_+^K ; \sum_k p_k = 1\right\}$ is the set of possible convex weights.

We note $x^\star \in \Delta^K$ the optimal allocation that maximizes F over Δ^K; the objective of the decision maker is to maximize the cumulated reward, or equivalently to minimize the regret $R(T)$, defined as the difference between the optimal reward $F(x^\star)$ and the average reward over $T \in \mathbb{N}^*$ stages (the fact that T is known in advance or not does not change much to the analysis (Degenne and Perchet, 2016)):

$$R(T) = F(x^\star) - \frac{1}{T} \sum_{t=1}^{T} \sum_{k=1}^{K} f_k(x_k^{(t)}) = \max_{x \in \Delta^K} F(x) - \frac{1}{T} \sum_{t=1}^{T} F(x^{(t)}).$$

The following diminishing return assumption on the reward functions f_k is natural and ensures that F is concave and continuous, ensuring the existence of $x^\star$.

Assumption 1 *The reward functions $f_k : [0,1] \to \mathbb{R}$ are concave, non-decreasing and $f_k(0) = 0$. Moreover we assume that they are differentiable, L-Lipschitz continuous and L'-smooth.*

This assumption means that the more the decision maker invest in a resource, the greater the revenue. Moreover, investing 0 gives nothing in return. Finally the marginal increase of revenue decreases.

We now describe the feedback model. At each time step the decision maker observes a noisy version of $\nabla F(x^{(t)})$, which is equivalent here to observing each $\nabla f_k(x_k^{(t)}) + \zeta_k^{(t)}$, where $\zeta_k^{(t)} \in \mathbb{R}$ is some white bounded noise. The assumption of noisy gradients is classical in stochastic optimization and is similarly relevant for our problem: this assumption is quite natural as the decision maker can evaluate, locally and with some noise, how much a small increase/decrease of an allocation $x_k^{(t)}$ affects the reward.

Consequently, the decision maker faces the problem of stochastic optimization of a concave and separable function over the simplex (yet with a cumulative regret minimization objective). Classical stochastic gradient methods from stochastic convex optimization would guarantee that the average regret decreases as $\widetilde{\mathcal{O}}\left((K/T)^{1/2}\right)$ in general and as $\widetilde{\mathcal{O}}\left(K/T\right)$ if the f_k are known to be strongly concave. However, even without strong concavity, we claim that it is possible to obtain better regret bounds than $\widetilde{\mathcal{O}}\left((K/T)^{1/2}\right)$ and, more importantly, to be adaptive to some complexity parameters.

The overarching objective is then to leverage the specific structure of this natural problem to provide a generic algorithm that is naturally **adaptive** to some complexity measure of the problem. It will, for instance, **interpolate** between the non-strongly concave and the strongly-concave rates without depending on the strong-concavity parameter, and **recover** the fast rate of classical multi-armed bandit (corresponding more or less to the case where the f_k functions are linear). Existing algorithms for adaptive stochastic convex optimization (Ramdas and Singh, 2013b; Iouditski and Nesterov, 2014) are not applicable in our case since they work for function-error minimization and not regret minimization (because of the prohibitively large stage lengths they are using).

2.2. The complexity class

As mentioned before, our algorithm will be adaptive to some general complexity parameter of the set of functions $\mathcal{F} = \{f_1, \dots, f_K\}$, which relies on the Łojasiewicz inequality (Bierstone and Milman, 1988; Bolte et al., 2010) that we state now, for concave functions (rather than convex).

Definition 1 *A function $f : \mathbb{R}^d \to \mathbb{R}$ satisfies the Łojasiewicz inequality with respect to $\beta \in [1, +\infty)$ on its domain $\mathcal{X} \subset \mathbb{R}^d$ if there exists a constant $c > 0$ such that*

$$\forall x \in \mathcal{X},\ \max_{x^* \in \mathcal{X}} f(x^*) - f(x) \leq c \|\nabla f(x)\|^\beta.$$

Given two functions $f, g : [0,1] \to \mathbb{R}$, we say that they satisfy **pair-wisely** the Łojasiewicz inequality with respect to $\beta \in [1, +\infty)$ if the function $(z \mapsto f(z) + g(x - z))$ satisfies the Łojasiewicz inequality on $[0, x]$ with respect to β for every $x \in [0,1]$.

It remains to define the finest class of complexity of a set of functions $\mathcal{F}$. It is defined with respect to binary trees, whose nodes and leaves are labeled by functions. The trees we consider are

constructed as follows. Starting from a finite binary tree of depth $\lceil \log_2(|\mathcal{F}|) \rceil$, its leaves are labeled with the different functions in $\mathcal{F}$ (and 0 for the remaining leaves if $|\mathcal{F}|$ is not a power of 2). The parent node of f_{left} and f_{right} is then labeled by the function $x \mapsto \max_{z \leq x} f_{\text{left}}(z) + f_{\text{right}}(x - z)$.

We say now that $\mathcal{F}$ satisfies **inductively** the Łojasiewicz inequality for $\beta \geq 1$ if in any binary tree labeled as above, any two siblings[3] satisfy pair-wisely the Łojasiewicz inequality for β.

Since the previous definition is quite intricate we can focus on some easier insightful sub-cases:

Uniformly concave functions (Iouditski and Nesterov, 2014) A function $f : \mathbb{R}^d \to \mathbb{R}$ is uniformly-concave with parameters $\rho \geq 2$ and $\mu > 0$ if and only if for all $x, y \in \mathbb{R}^d$ and for all $\alpha \in [0, 1]$,

$$f(\alpha x + (1-\alpha)y) \geq \alpha f(x) + (1-\alpha) f(y) + \frac{\mu}{2} \alpha (1-\alpha) \left[\alpha^{\rho - 1} + (1-\alpha)^{\rho - 1} \right] \|x - y\|^{\rho}.$$

If all functions f_k are (ρ_k, μ_k)-uniformly convex, then the relevant complexity parameter (for the rate of convergence) is $\beta_{\mathcal{F}} = \frac{\rho_{\mathcal{F}}}{\rho_{\mathcal{F}} - 1}$ where $\rho_{\mathcal{F}} := \max_k \rho_k$.

Tsybakov Noise Condition (TNC) (Ramdas and Singh, 2013b) A function $f : \mathbb{R}^d \to \mathbb{R}$ satisfies the global TNC if with parameters $\kappa \geq 2$ and $\mu > 0$ if and only if for all $x, y \in \mathbb{R}^d$,

$$|f(x) - f(y)| \geq \mu \|x - y\|^{\kappa}.$$

If all functions f_k satisfies (κ_k, μ_k)-TNC, then the relevant complexity parameter (for the rate of convergence) is $\beta_{\mathcal{F}} = \frac{\kappa_{\mathcal{F}}}{\kappa_{\mathcal{F}} - 1}$ where $\kappa_{\mathcal{F}} := \max_k \kappa_k$.

More details about the Łojasiewicz inequality (as well as examples and counter-examples) and its links with uniform convexity can be found in Appendix A. Additional examples of class of functions satisfying inductively the Łojasiewicz inequality can be found in Appendix B.

One could ask why the class of Łojasiewicz functions is interesting. A result of Łojasiewicz (1965) shows that all analytic functions satisfy the (local) Łojasiewicz inequality with a parameter $\beta > 1$. This is a strong result motivating our interest for the class of functions satisfying the Łojasiewicz inequality. More precisely we prove the following proposition in Appendix B.

Proposition 2 *If the functions $\{f_1, \dots, f_K\}$ are real analytic and strictly concave then the class $\mathcal{F}$ satisfy inductively the Łojasiewicz inequality with a parameter $\beta_{\mathcal{F}} > 1$.*

In the following section, we introduce a **generic, parameter free algorithm** that is **adaptive to the complexity $\beta_{\mathcal{F}} \in [1, +\infty)$ of the problem**. Note that $\beta_{\mathcal{F}}$ is not necessarily known by the agent and therefore the fact that the algorithm is adaptive to the parameter is particularly interesting. The simplest case $K = 2$ provides many insights and will be used as a sub-routine for more resources. Therefore, we will first focus on this case.

3. Stochastic Gradient Feedback for $K = 2$

We first focus on only $K = 2$ resources. In this case, we rewrite the reward function F as

$$F(x) = f_1(x_1) + f_2(x_2) = f_1(x_1) + f_2(1 - x_1).$$

For the sake of clarity we simply note $x = x_1$ and we define $g(x) \triangleq F(x) - F(x^\star)$. Note that $g(x^\star) = 0$ and that g is a non-positive concave function. Using these notations, at each time step t the agent chooses $x^{(t)} \in [0, 1]$, suffers $|g(x^{(t)})|$ and observes $g'(x^{(t)}) + \varepsilon_t$ where $\varepsilon_t \in [-1, 1]$ i.i.d.

3. To be precise, we could only require that this property holds for any siblings that are not children of the root. For those two, we only need that the mapping $f_{\text{left}}(z) + f_{\text{right}}(1 - z)$ satisfies the local Łojasiewicz inequality.

3.1. Description of the main algorithm

The basic algorithm we follow to optimize g is a binary search. Each query point x (for example $x = 1/2$) is sampled repeatedly and sufficiently enough (as long as 0 belongs to some confidence interval) to guarantee that the sign of $g'(x)$ is known with arbitrarily high probability, at least $1 - \delta$.

Algorithm 1 Binary search algorithm

Require: T time horizon, δ confidence parameter
Search interval $I_0 \leftarrow [0,1]$; $t \leftarrow 1$; $j \leftarrow 1$
while $t \leq T$ **do**
  $x_j \leftarrow \text{center}(I_{j-1})$; $S_j \leftarrow 0$; $N_j \leftarrow 0$
  while $0 \in \left[\frac{S_j}{N_j} \pm \sqrt{\frac{2\log(\frac{2T}{\delta})}{N_j}}\right]$ **do**
    Sample x_j and get X_t, noisy value of $\nabla g(x_j)$
    $S \leftarrow S_j + X_t$, $N_j \leftarrow N_j + 1$
  end while
  if $\frac{S_j}{N_j} > \sqrt{\frac{2\log(\frac{2T}{\delta})}{N_j}}$ **then**
    $I_j \leftarrow [x_j, \max(I_{j-1})]$
  else
    $I_j \leftarrow [\min(I_{j-1}), x_j]$
  end if
  $t \leftarrow t + N_j$; $j \leftarrow j + 1$
end while
return x_j

Algorithm 1 is not conceptually difficult (but its detailed analysis of performances is however): it is just a binary search where each query point is sampled enough time to be sure on which "direction" the search should proceed next. Indeed, because of the concavity and monotone assumptions on f_1 and f_2, if $x < x^\star$ then

$$x < x^\star \iff \nabla g(x) = \nabla f_1(x) - \nabla f_2(1-x) < 0 \,.$$

By getting enough noisy samples of $\nabla g(x)$, it is possible to decide, based on its sign, whether $x^\star$ lies on the right or the left of x. If x_j is the j-th point queried by the binary search (and letting $j_{\max}$ be the total number of different queries), we get that the binary search is successful with high probability, i.e., that with probability at least $1 - \delta T$ for each $j \in \{1, \dots, j_{\max}\}$, $|x_j - x^\star| \leq 2^{-j}$. We also call N_j the actual number of samples of x_j which is bounded by $8\log(2T/\delta)/|g'(x_j)^2|$ by Lemma 3, whose proof can be found in Appendix E.

Lemma 3 *Let $x \in [-1,1]$ and $\delta \in (0,1)$. For any random variable $X \in [x-1, x+1]$ of expectation x, at most $N_x = \dfrac{8}{x^2}\log(2T/\delta)$ i.i.d. samples $X_1, X_2, \dots, X_n$ are needed to figure out the sign of x with probability at least $1 - \delta$. Indeed, one just need stop sampling as soon as*

$$0 \notin \left[\frac{1}{n}\sum_{t=1}^{n} X_t \pm \sqrt{\frac{2\log(2T/\delta)}{n}}\right]$$

and determine the sign of x is positive if $\frac{1}{n}\sum_{t=1}^{n} X_t \geq \sqrt{\frac{2\log(2T/\delta)}{n}}$ and negative otherwise.

The regret of the algorithm then rewrites as

$$R(T) = \frac{1}{T}\sum_{t=1}^{T}|g(x^{(t)})| = \frac{1}{T}\sum_{j=1}^{j_{\max}} N_j|g(x_j)| \leq \frac{8}{T}\log(2T/\delta)\sum_{j=1}^{j_{\max}}\frac{|g(x_j)|}{g'(x_j)^2}. \tag{1}$$

Our analysis of the algorithm performances are based on the control of the last sum in Equation (1).

3.2. Strongly concave functions

First, we consider the case where the functions f_1 and f_2 are strongly concave.

Theorem 4 *If the algorithm is run with $\delta = 2/T^2$ and if g is a L'-smooth and α-strongly concave function on $[0,1]$, then there exists a universal positive constant κ such that*

$$\mathbb{E}R(T) \leq \frac{\kappa}{\alpha}\frac{\log(T)}{T}.$$

This results shows that our algorithm reaches the same rates as the stochastic gradient descent in the smooth and strongly concave case. The proof is delayed to Appendix C for the sake of fluency.

3.3. Analysis in the non-strongly concave case

We now consider the case where g is only concave, without being necessarily strongly concave.

Theorem 5 *Assume that g satisfies the local Łojasiewicz inequality w.r.t. $\beta \geq 1$ and $c > 0$ and that the algorithm is run with $\delta = 2/T^2$. Then there exists a universal constant $\kappa > 0$ such that*

in the case where $\beta > 2$, $\mathbb{E}[R(T)] \leq \kappa\frac{c^{2/\beta}L^{1-2/\beta}}{1-2^{2/\beta-1}}\frac{\log(T)}{T}$;

in the case where $\beta \leq 2$, $\mathbb{E}[R(T)] \leq \kappa \cdot c\left(\frac{\log(T)^2}{T}\right)^{\beta/2}$.

The proof of Theorem 5 relies on bouding the sum in Equation (1), which can be recast as a constrained minimization problem. It is postponed to Appendix C for clarity reasons.

3.4. Lower bounds

We now provide a lower bound for our problem that indicates that our rates of convergence are optimal up to $\mathrm{poly}(\log(T))$ terms. For $\beta \geq 2$, it is trivial to see that no algorithm can have a regret smaller than $\Omega(1/T)$, hence we shall focus on $\beta \in [1,2]$.

Theorem 6 *Given the horizon T fixed, for any algorithm, there exists a pair of functions f_1 and f_2 that are concave, non-decreasing and such that $f_i(0) = 0$, such that*

$$\mathbb{E}R(T) \geq c_\beta T^{-\frac{\beta}{2}}$$

where $c_\beta > 0$ is some constant independent of T.

The proof and arguments are rather classical now (Shamir, 2013; Bach and Perchet, 2016): we exhibit two pairs of functions whose gradients are $1/\sqrt{T}$-close with respect to the uniform norm. As no algorithm can distinguish between them with arbitrarily high probability, the regret will scale more or less as the difference between those functions which is as expected of the order of $T^{-\beta/2}$. More details can be found in Appendix C.

3.5. The specific case of linear (or dominating) resources - the Multi-Armed Bandit case

We focus in this section on the specific case where the resources have linear efficiency, meaning that $f_i(x) = \alpha_i x$ for some unknown parameter $\alpha_i \geq 0$. In that case, the optimal allocation of resource consists in putting all the weights to the resource with the highest parameter α_i.

More generally, if $f_1'(1) \geq f_2'(0)$, then one can easily check that the optimal allocation consists in putting again all the weight to the first resource (and, actually, the converse statement is also true).

It happens that in this specific case, the learning is fast as it can be seen as a particular instance of Theorem 5 in the case where $\beta > 2$. Indeed, let us assume that $\operatorname{argmax}_{x\in\mathbb{R}} g(x) > 1$, meaning that $\max_{x\in[0,1]} g(x) = g(1)$, so that, by concavity of g it holds that $g'(x) \geq g'(1) > 0$ thus g is increasing on $[0,1]$. In particular, this implies that for every $\beta > 2$:

$$\forall x \in [0,1],\ g(1) - g(x) = |g(x)| \leq g(0) \leq \frac{g(0)}{g'(1)^\beta} g'(1)^\beta \leq \frac{g(0)}{g'(1)^\beta} g'(x)^\beta = c|g'(x)|^\beta,$$

showing that g verifies the Łojasiewicz inequality for every $\beta > 2$ and with constant $c = g(0)/g'(1)^\beta$. As a consequence, Theorem 5 applies and we obtain fast rates of convergence in $\mathcal{O}\left(\log(T)/T\right)$.

However, we propose in the following an alternative analysis of the algorithm for that specific case. Recall that regret can be bounded as

$$R(T) = \frac{8}{T}\log(2T/\delta) \sum_{j=1}^{j_{\max}} \frac{|g(x_j)|}{g'(x_j)^2} = \frac{8}{T}\log(2T/\delta) \sum_{j=1}^{j_{\max}} \frac{|g(1-1/2^j)|}{g'(1-1/2^j)^2}.$$

We now notice that

$$\left|g\left(1-2^{-j}\right)\right| = g(1) - g\left(1-2^{-j}\right) = \int_{1-1/2^j}^{1} g'(x)\,\mathrm{d}x \leq 2^{-j} g'\left(1-2^{-j}\right).$$

And finally we obtain the following bound on the regret:

$$R(T) \leq \frac{8}{T}\log(2T/\delta) \sum_{j=1}^{j_{\max}} \frac{1}{2^j}\frac{1}{g'(1)} \leq \frac{8}{T}\frac{\log(2T/\delta)}{g'(1)} \leq \frac{24}{\Delta}\frac{\log(T)}{T}$$

since $g'(1-1/2^j) > g'(1)$ and with the choice of $\delta = 2/T^2$. We have noted $\Delta \triangleq g'(1)$ in order to enlighten the similarity with the multi-armed bandit problems with 2 arms. We have indeed $g'(1) = f_1'(1) - f_2'(0) > 0$ which can be seen as the gap between both arms. It is especially true in the linear case where $f_i(x) = \alpha_i x$ as $\Delta = |\alpha_1 - \alpha_2|$ and the gap between arms is by definition of the multi-armed bandit problem $|f(1) - f(0)| = |\alpha_1 - \alpha_2|$.

4. Stochastic gradient feedback and $K \geq 3$ resources

We now consider the case with more than 2 resources. The generic algorithm still relies on binary searches as in the previous section with $K = 2$ resources, but we have to imbricate them in a tree-like structure to be able to leverage the Łojasiewicz inequality assumption. The goal of this section is to present our algorithm and to prove the following theorem, which is a generalization of Theorem 5.

Theorem 7 *Assume that $\mathcal{F} = \{f_1, f_2, \ldots, f_k\}$ satisfies inductively the Łojasiewicz inequality w.r.t. the parameters $\beta_{\mathcal{F}} \geq 1$ and $c > 0$. Then there exists a universal constant $\kappa > 0$ such that our algorithm, run with with $\delta = 2/T^2$, ensures*

in the case $\beta_{\mathcal{F}} > 2$, *then* $\mathbb{E}[R(T)] \leq \kappa \dfrac{c^{2/\beta_{\mathcal{F}}} L^{1-2/\beta_{\mathcal{F}}}}{1 - 2^{2/\beta_{\mathcal{F}}-1}} K \dfrac{\log(T)^{\log_2(K)}}{T}$;

in the case $\beta_{\mathcal{F}} \leq 2$, *then* $\mathbb{E}[R(T)] \leq \kappa \cdot cK \left(\dfrac{\log(T)^{\log_2(K)+1}}{T} \right)^{\beta_{\mathcal{F}}/2}$.

Let us first mention why the following natural extension of the algorithm for $K = 2$ does not work. Assume that the algorithm would sample repeatedly a point $x \in \Delta^K$ until the different confidence intervals around the gradient $\nabla f_k(x_k)$ do not overlap. When this happens with only 2 resources, then it is known that the optimal $x^\star$ allocates more weight to the resource with the highest gradient and less weight to the resource with the lowest gradient. This property only holds partially for $K \geq 3$ resources. Given $x \in \Delta^K$, even if we have a (perfect) ranking of gradient $\nabla f_1(x_1) > \ldots > \nabla f_K(x_K)$ we can only infer that $x_1^\star \geq x_1$ and $x_K^\star \leq x_K$. For intermediate gradients we cannot (without additional assumptions) infer the relative position of $x_j^\star$ and x_j.

To circumvent this issue, we are going to build a binary tree, whose leaves are labeled arbitrarily from $\{f_1, \ldots, f_K\}$ and we are going to run inductively the algorithm for $K = 2$ resources at each node, *i.e.,* between its children f_{left} and f_{right}. The main difficulty is that we no longer have unbiased samples of the gradients of those functions (but only those located at the leaves).

4.1. Insights on the main algorithm

To be more precise, recall we aim at maximizing the mapping (and controlling the regret)

$$F(x) = \sum_{k=1}^{K} f_k(x_k) \quad \text{with } x = (x_1, \ldots, x_K) \in \Delta^K.$$

As we have a working procedure to handle only $K = 2$ resources, we will adopt a divide-and-conquer strategy by diving the mapping F into two sub-mapping $F_1^{(1)}$ and $F_2^{(1)}$ defined by

$$F_1^{(1)}(x) = \sum_{k=1}^{\lceil K/2 \rceil} f_k(x_k) \quad \text{and} \quad F_2^{(1)}(x) = \sum_{k=\lceil K/2 \rceil + 1}^{K} f_k(x_k).$$

Since the original mapping F is separable, we can reduce the optimization of F over the simplex Δ^K to the optimization of a sum of two functions over the simplex of dimension 1 (thus going back

to the case of $K = 2$ resources). Indeed,

$$\max_{\|x\|_1=1} F(x) = \max_{z\in[0,1]} \left(\max_{\|x\|_1=z} F_1^{(1)}(x) + \max_{\|x\|_1=1-z} F_2^{(1)}(x) \right) \\ \triangleq \max_{z\in[0,1]} H_1^{(1)}(z) + H_2^{(1)}(1-z) \, .$$

Now we aim to apply the machinery of $K = 2$ resources to the reward mappings $H_1^{(1)}$ and $H_2^{(1)}$. The major issue is that we do not have directly access to the gradients $\nabla H_1^{(1)}(z)$ and $\nabla H_2^{(1)}(1-z)$ of those functions because they are defined via an optimization problem. However, can apply again the divide-and-conquer approach to $H_1^{(1)}$ and compute its gradient using the envelope theorem (Afriat, 1971). Indeed, divide again $F_1^{(1)}$ into the two following mappings $F_1^{(2)}$ and $F_2^{(2)}$ defined by

$$F_1^{(2)}(x) = \sum_{k=1}^{\lceil K/4 \rceil} f_k(x_k) \quad \text{and} \quad F_2^{(2)}(x) = \sum_{k=\lceil K/4 \rceil+1}^{\lceil K/2 \rceil} f_k(x_k).$$

Then as above, we can rewrite the optimization problem defining $H_1^{(1)}$ as another optimization problem over $[0, z]$ by noting that

$$H_1^{(1)}(z) = \max_{\|x\|_1=z} F_1^{(1)}(x) = \max_{\omega\in[0,z]} \left(\max_{\|x\|_1=\omega} F_1^{(2)}(x) + \max_{\|x\|_1=z-\omega} F_2^{(2)}(x) \right) \\ \triangleq \max_{\omega\in[0,z]} H_1^{(2)}(\omega) + H_2^{(2)}(z-\omega) \, .$$

The envelope theorem now gives the following lemma (whose proof is immediate and omitted).

Lemma 8 *Let $\omega_z^* \in [0, z]$ be the maximizer of $H_1^{(2)}(\omega) + H_2^{(2)}(z-\omega)$, then*

$$\nabla H_1^{(1)}(z) = \begin{cases} \nabla H_1^{(2)}(\omega_z^*) = \nabla H_2^{(2)}(z-\omega_z^*) & \text{if } \omega_z^* \in (0, z) \\ \nabla H_2^{(2)}(z) & \text{if } \omega_z^* = 0 \\ \nabla H_1^{(2)}(z) & \text{if } \omega_z^* = z \end{cases} .$$

Recall that gradients of $H_1^{(1)}(z)$ and $H_2^{(1)}(1-z)$ were needed to apply the $K = 2$ machinery to the optimization of F once this problem is rewritten as $\max_z H_1^{(1)}(z) + H_2^{(1)}(1-z)$. Lemma 8 provides them, as the gradient of yet other functions $H_1^{(2)}$ and/or $H_2^{(2)}$. Notice that if $K = 4$, then those two functions are actually the two basis functions f_1 and f_2, so the agent has direct access to their gradient (up to some noise). It only remains to find the point ω_z^* which is done with the binary search introduced in the previous section.

If $K > 4$, the gradient of $H_1^{(2)}$ (and, of course, of $H_2^{(2)}$) is not directly accessible, but we can again divide $H_1^{(2)}$ into two other functions $H_1^{(3)}$ and $H_2^{(3)}$. Then the gradient of $H_1^{(2)}$ will be expressed, via Lemma 8, as gradients of $H_1^{(3)}$ and/or $H_2^{(3)}$ at some specific point (again, found by binary searches as in $K = 2$). We can repeat this process as long as $H_1^{(k)}$ and $H_2^{(k)}$ are not basis functions in $\mathcal{F}$ and $\mathcal{F}$ can be "divided" to compute recursively the gradients of each $H_j^{(k)}$ up to $H_1^{(1)}$ and $H_2^{(1)}$, up to the noise and some estimation errors that must be controlled.

4.2. The Generic Algorithm

A more detailed version of our generic algorithm, with the notations required for the proof can be found in Appendix D. To give some intuitions, consider a binary tree whose root is labeled by the function $F(x) = \sum_{k=1}^{K} f_k(x_k) \triangleq F_1^{(0)}(x)$ that we want to maximize. We are going to label recursively the nodes of this tree by functions and the leaves of this tree are going to be the elements of $\mathcal{F}$. Denote by $F_j^{(i)}$ the function created at the nodes of depth i, with j an increasing index from the left to the right of the tree. If $F_j^{(i)}(x)$ is not a leaf, *i.e.,* not an element of $\mathcal{F}$, then there exist two indices $k_1 < k_2$ such that $F_j^{(i)}(x) = \sum_{k=k_1}^{k_2} f_k(x_k)$ and we define

$$F_{2j-1}^{(i+1)}(x) = \sum_{k=k_1}^{\lfloor (k_1+k_2)/2 \rfloor} f_k(x_k) \quad \text{and} \quad F_{2j}^{(i+1)}(x) = \sum_{k=\lfloor (k_1+k_2)/2 \rfloor + 1}^{k_2} f_k(x_k).$$

If K is not a power of 2 we can add artificial functions with value 0 in order to obtain a balanced tree. The optimization of $F_j^{(i)}$ can be done recursively since

$$\max_{\|x\|_1 = z_n} F_j^{(i)}(x) = \max_{z_{n+1} \in [0, z_n]} \left(\max_{\|x\|_1 = z_{n+1}} F_{2j-1}^{(i+1)}(x) + \max_{\|x\|_1 = z_n - z_{n+1}} F_{2j}^{(i+1)}(x) \right).$$

In order to clarify notations we also define the following mappings as in Section 4.1:

Definition 9 *For every i and j in the constructed binary tree of functions, we define*

$$H_j^{(i)}(z) \triangleq \max_{\|x\|_1 = z} F_j^{(i)}(x) \quad \textit{and} \quad G_j^{(i)}(z; y) \triangleq H_{2j-1}^{(i+1)}(z) + H_{2j}^{(i+1)}(y - z).$$

We also note $\mathcal{D}_j^{(i)}(v)$ the binary search whose goal is to optimize the function $G_j^{(i)}(\cdot\,; v)$. With these notations, it holds that for all $z_n \in [0, 1]$,

$$H_j^{(i)}(z_n) = \max_{z_{n+1} \in [0, z_n]} G_j^{(i)}(z_{n+1}; z_n) = \max_{z_{n+1} \in [0, z_n]} H_{2j-1}^{(i+1)}(z_{n+1}) + H_{2j}^{(i+1)}(z_n - z_{n+1}),$$

and the gradient of $H_j^{(i)}(z)$ can be expressed in terms of those of $H_{2j-1}^{(i+1)}$ and $H_{2j}^{(i+1)}$ by Lemma 8.

As a consequence, the gradients of $H_j^{(i)}$ can be **recursively** approximated using estimates of the gradients of their children (in the binary tree). Indeed, assume that one has access to ε-approximations of $\nabla H_{2j-1}^{(i+1)}$ and $\nabla H_{2j}^{(i+1)}$. Then Lemma 8 directly implies that a ε-approximation of its gradient $\nabla H_j^{(i)}(z)$ can be computed by a binary search on $[0, z]$. Moreover, notice that if a binary search is optimizing $H_j^{(i)}$ on $[0, z]$ and is currently querying the point ω, then the level of approximation required (and automatically set to) is equal to $|\nabla H_{2j-1}^{(i+1)}(\omega) - \nabla H_{2j}^{(i+1)}(z - \omega)|$. This is the crucial property that allows a control on the regret.

The main algorithm can now be simply summarized as performing a binary search for the maximization of $H_1^{(1)}(z) + H_2^{(1)}(1 - z)$ using recursive estimates of $\nabla H_1^{(1)}$ and $\nabla H_2^{(1)}$.

4.3. Main ideas of the proof of Theorem 7

The full analysis of Theorem 7 is too involved to be detailed thoroughly here. The detailed proof is provided in Appendix D. We provide nevertheless a very natural intuition in the case of strongly concave mappings or $\beta > 2$, as well as the main ingredients of the general proof.

Recall that in the case where $\beta > 2$, the average regret of the algorithm for $K = 2$ scales as $\log(T)/T$. As a consequence, running a binary search induces a cumulative regret of the order of $\log(T)$. The generic algorithm is defined recursively over a binary tree of depth $\log_2(K)$ and each function in the tree is defined by a binary search over its children. So at the end, to perform a binary search over $H_1^{(1)}(z) + H_2^{(1)}(1-z)$, the algorithm imbricates $\log_2(K)$ binary searches to compute gradients. The error made by these binary searches cumulate (multiplicatively) ending up in a cumulative regret term of the order of $\log(T)^{\log_2(K)}$.

For $\beta < 2$, the analysis is more intricate, but the main idea is the same one; to compute a gradient, $\log_2(K)$ binary searches must be imbricated and their errors cumulate to give Theorem 7. We give here some of the main ingredients of the proof. As explained in Appendix D we can associate a regret for each binary search, and we call $R_j^{(i)}(v)$ the regret associated to the binary search $\mathcal{D}_j^{(i)}(v)$. Since we have more than 2 resources we have to imbricate the binary searches in a recursive manner in order to get access to the gradients of the functions $H_j^{(i)}$. This will lead to a regret $R_j^{(i)}(v)$ for the binary search $\mathcal{D}_j^{(i)}(v)$ that will recursively depend on the regrets of the binary searches corresponding to the children (in the tree) of $\mathcal{D}_j^{(i)}(v)$. An important part of the proof of Theorem 7 is therefore devoted to proving the following proposition.

Proposition 10 *The regret $R_j^{(i)}(v)$ of the binary search $\mathcal{D}_j^{(i)}(v)$ is bounded by:*

$$R_j^{(i)}(v) \leq \sum_{r=1}^{r_{\max}} 8\log(2T/\delta)\frac{\left|g_j^{(i)}(w_r;v)\right|}{\left|\nabla g_j^{(i)}(w_r;v)\right|^2}\log(T)^{\log_2(K)-1-i} + R_{2j-1}^{(i+1)}(w_r) + R_{2j}^{(i+1)}(v-w_r),$$

where $\{w_1,\dots,w_{r_{\max}}\}$ are the different samples of $\mathcal{D}_j^{(i)}(v)$ and $g_j^{(i)}(\cdot\,;v) \doteq G_j^{(i)}(\cdot\,;v) - \max_z G_j^{(i)}(z;v)$.

The goal of the remaining of the proof of Theorem 7 is to bound $R_1^{(0)}(1)$. The very natural way to do it is to use the previous proposition with the Łojasiewicz inequality to obtain a simple recurrence relation between the successive values of $R_j^{(i)}$. The end of the proof is then similar to the proofs done in the case $K = 2$. Besides we can note that the statement of Proposition 10 shows clearly that adding more levels to the tree results in an increase of the exponent of the $\log(T)$ factor.

5. Conclusion

We have considered the problem of multi-resource allocation under the classical assumption of diminishing returns. This appears to be a concave optimization problem and we proposed an algorithm based on imbricated binary searches to solve it. Our algorithm is particularly interesting in the sense that it is fully adaptive to all parameters of the problem (strong convexity, smoothness, Łojasiewicz exponent, etc.). Our analysis provides meaningful upper bound for the regret that matches the lower bounds, up to logarithmic factors. The experiments we conducted (see Appendix F) validate as expected the theoretical guarantees of our algorithm, as empirically regret seems to decrease polynomially with T with the right exponent.

Acknowledgments

X. Fontaine was supported by grants from Région Ile-de-France. This work was supported by a public grant as part of the Investissement d'avenir project, reference ANR-11-LABX-0056-LMH, LabEx LMH, in a joint call with Gaspard Monge Program for optimization, operations research and their interactions with data sciences. V. Perchet also acknowledges the support of the ANR under the grant ANR-19-CE23-0026.

References

Yasin Abbasi-Yadkori, Dávid Pál, and Csaba Szepesvári. Improved algorithms for linear stochastic bandits. In *Proceedings of the 24th International Conference on Neural Information Processing Systems*, NIPS'11, pages 2312–2320, USA, 2011. Curran Associates Inc. ISBN 978-1-61839-599-3. URL http://dl.acm.org/citation.cfm?id=2986459.2986717.

SN Afriat. Theory of maxima and the method of lagrange. *SIAM Journal on Applied Mathematics*, 20(3):343–357, 1971.

Alekh Agarwal, Dean P Foster, Daniel J Hsu, Sham M Kakade, and Alexander Rakhlin. Stochastic convex optimization with bandit feedback. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 24*, pages 1035–1043. Curran Associates, Inc., 2011. URL http://papers.nips.cc/paper/4475-stochastic-convex-optimization-with-bandit-feedback.pdf.

Shipra Agrawal and Nikhil R. Devanur. Bandits with concave rewards and convex knapsacks. In *Proceedings of the Fifteenth ACM Conference on Economics and Computation*, EC '14, pages 989–1006, New York, NY, USA, 2014. ACM. ISBN 978-1-4503-2565-3. doi: 10.1145/2600057.2602844. URL http://doi.acm.org/10.1145/2600057.2602844.

Shipra Agrawal and Nikhil R. Devanur. Fast algorithms for online stochastic convex programming. In *Proceedings of the Twenty-sixth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '15, pages 1405–1424, Philadelphia, PA, USA, 2015. Society for Industrial and Applied Mathematics. URL http://dl.acm.org/citation.cfm?id=2722129.2722222.

Francis Bach and Vianney Perchet. Highly-smooth zero-th order online optimization. In Vitaly Feldman, Alexander Rakhlin, and Ohad Shamir, editors, *29th Annual Conference on Learning Theory*, volume 49 of *Proceedings of Machine Learning Research*, pages 257–283, Columbia University, New York, New York, USA, 23–26 Jun 2016. PMLR.

Melvyn S Berger. *Nonlinearity and functional analysis: lectures on nonlinear problems in mathematical analysis*, volume 74. Academic press, 1977.

Quentin Berthet and Vianney Perchet. Fast rates for bandit optimization with upper-confidence frank-wolfe. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 2225–2234. 2017.

Edward Bierstone and Pierre Milman. Semianalytic and subanalytic sets. *Publications Mathématiques de l'IHÉS*, 67:5–42, 1988. URL http://www.numdam.org/item/PMIHES_1988__67__5_0.

Jerome Bolte, Aris Daniilidis, Olivier Ley, and Laurent Mazet. Characterizations of lojasiewocz inequalities: Subgradient flows, talweg, convexity. *Transactions of the American Mathematical Society*, 362(6):3319–3363, 2010.

Sébastien Bubeck and Nicolò Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1):1–122, 2012. ISSN 1935-8237. doi: 10.1561/2200000024. URL http://dx.doi.org/10.1561/2200000024.

Marat Valievich Burnashev and Kamil'Shamil'evich Zigangirov. An interval estimation problem for controlled observations. *Problemy Peredachi Informatsii*, 10(3):51–61, 1974.

Rui M Castro and Robert D Nowak. Minimax bounds for active learning. *IEEE Transactions on Information Theory*, 54(5):2339–2353, 2008.

J. M. Colom. The resource allocation problem in flexible manufacturing systems. In Wil M. P. van der Aalst and Eike Best, editors, *Applications and Theory of Petri Nets 2003*, pages 23–35, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

Yuval Dagan and Koby Crammer. A better resource allocation algorithm with semi-bandit feedback. In Firdaus Janoos, Mehryar Mohri, and Karthik Sridharan, editors, *Proceedings of Algorithmic Learning Theory*, volume 83 of *Proceedings of Machine Learning Research*, pages 268–320. PMLR, 07–09 Apr 2018. URL http://proceedings.mlr.press/v83/dagan18a.html.

Varsha Dani, Thomas P. Hayes, and Sham M. Kakade. Stochastic linear optimization under bandit feedback. In *COLT*, 2008.

Rémy Degenne and Vianney Perchet. Anytime optimal algorithms in stochastic multi-armed bandits. In Maria Florina Balcan and Kilian Q. Weinberger, editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1587–1595, New York, New York, USA, 20–22 Jun 2016. PMLR.

Nikhil R Devanur, Kamal Jain, Balasubramanian Sivan, and Christopher A Wilkens. Near optimal online algorithms and fast approximation algorithms for resource allocation problems. *Journal of the ACM (JACM)*, 66(1):7, 2019.

O Gross. A class of discrete-type minimization problems. Technical report, RAND CORP SANTA MONICA CA, 1956.

Anatoli Iouditski and Yuri Nesterov. Primal-dual subgradient methods for minimizing uniformly convex functions. *arXiv preprint arXiv:1401.1792*, 2014.

Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the polyak-łojasiewicz condition. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 795–811. Springer, 2016.

Naoki Katoh and Toshihide Ibaraki. Resource allocation problems. In *Handbook of combinatorial optimization*, pages 905–1006. Springer, 1998.

Bernard O Koopman. The optimum distribution of effort. *Journal of the Operations Research Society of America*, 1(2):52–63, 1953.

Nitish Korula, Vahab Mirrokni, and Morteza Zadimoghaddam. Online submodular welfare maximization: Greedy beats 1/2 in random order. *SIAM Journal on Computing*, 47(3):1056–1086, 2018.

Tor Lattimore, Koby Crammer, and Csaba Szepesvari. Linear multi-resource allocation with semi-bandit feedback. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 964–972. Curran Associates, Inc., 2015. URL http://papers.nips.cc/paper/5931-linear-multi-resource-allocation-with-semi-bandit-feedback.pdf.

Stanislaw Łojasiewicz. Ensembles semi-analytiques. preprint, IHES, July 1965.

Aaditya Ramdas and Aarti Singh. Optimal rates for first-order stochastic convex optimization under tsybakov noise condition. In *Proceedings of the 30th International Conference on International Conference on Machine Learning*, 2013a.

Aaditya Ramdas and Aarti Singh. Algorithmic connections between active learning and stochastic convex optimization. In *International Conference on Algorithmic Learning Theory*, pages 339–353. Springer, 2013b.

Mohsen Amini Salehi, Jay Smith, Anthony A Maciejewski, Howard Jay Siegel, Edwin KP Chong, Jonathan Apodaca, Luis D Briceno, Timothy Renner, Vladimir Shestak, Joshua Ladd, et al. Stochastic-based robust dynamic resource allocation for independent tasks in a heterogeneous computing system. *Journal of Parallel and Distributed Computing*, 97:96–111, 2016.

P.A. Samuelson and W.D. Nordhaus. *Macroeconomics*. McGraw-Hill international editions. Irwin McGraw-Hill, 2005. ISBN 9780071111881. URL https://books.google.co.il/books?id=3S9gPgAACAAJ.

Ohad Shamir. On the complexity of bandit and derivative-free stochastic convex optimization. In Shai Shalev-Shwartz and Ingo Steinwart, editors, *Proceedings of the 26th Annual Conference on Learning Theory*, volume 30 of *Proceedings of Machine Learning Research*, pages 3–24, Princeton, NJ, USA, 12–14 Jun 2013. PMLR.

Adam Smith. *An Inquiry into the Nature and Causes of the Wealth of Nations*. McMaster University Archive for the History of Economic Thought, 1776. URL https://EconPapers.repec.org/RePEc:hay:hetboo:smith1776.

Haijun Zhang, Fang Fang, Julian Cheng, Keping Long, Wei Wang, and Victor CM Leung. Energy-efficient resource allocation in noma heterogeneous networks. *IEEE Wireless Communications*, 25(2):48–53, 2018.

Appendix A. Additional results on the Łojasiewicz inequality

We give here more detailed about the Łojasiewicz inequality. In this section we state all the results for convex functions. Their equivalents for concave functions are easily obtained by symmetry. The first one is the fact that every uniformly convex function verifies the Łojasiewicz inequality.

Definition 11 *A function $f : \mathbb{R}^d \to \mathbb{R}$ satisfies the Łojasiewicz inequality if*

$$\forall x \in \mathcal{X},\ f(x) - \min_{x^* \in \mathcal{X}} f(x^*) \leq \mu \|\nabla f(x)\|^{\beta}.$$

Definition 12 *A function $f : \mathbb{R}^d \to \mathbb{R}$ is uniformly-convex with parameters $\rho \geq 2$ and $\mu > 0$ if and only if for all $x, y \in \mathbb{R}^d$ and for all $\alpha \in [0, 1]$,*

$$f(\alpha x + (1-\alpha)y) \leq \alpha f(x) + (1-\alpha) f(y) - \frac{\mu}{2}\alpha(1-\alpha)\left[\alpha^{\rho-1} + (1-\alpha)^{\rho-1}\right] \|x - y\|^{\rho}.$$

Proposition 13 *If f is a differentiable (ρ, μ)-uniformly convex function then it satisfies the Łojasiewicz inequality with parameters $\beta = \rho/(\rho-1)$ and $c = \left(\frac{2}{\mu}\right)^{1/(\rho-1)} \frac{\rho - 1}{\rho^{\rho/(\rho-1)}}$.*

Proof A characterization of differentiable uniformly convex function (see for example (Iouditski and Nesterov, 2014)) gives that for all $x, y \in \mathbb{R}^d$

$$f(y) \geq f(x) + \langle \nabla f(x), y - x \rangle + \frac{1}{2}\mu \|x - y\|^{\rho}.$$

Consequently, noting $f(x^*) = \inf f(x)$,

$$f(x^*) \geq \inf_y \underbrace{\left\{ f(x) + \langle \nabla f(x), y - x \rangle + \frac{1}{2}\mu \|x - y\|^{\rho} \right\}}_{g(y)}.$$

We now want to minimize the function g which is a strictly convex function. We have

$$\nabla g(y) = \nabla f(x) + \frac{\mu}{2}\rho \|x - y\|^{\rho-2} (y - x).$$

g reaches its minimum for $\nabla g(y) = 0$ and $\nabla f(x) = -\frac{\mu}{2}\rho \|x - y\|^{\rho-2} (y - x)$. This gives

$$f(x^*) \geq f(x) + \frac{\mu}{2} \|x - y\|^{\rho} (1 - \rho).$$

Since $\|\nabla f(x)\| = \frac{\mu\rho}{2} \|x - y\|^{\rho-1}$ we obtain

$$\begin{aligned} f(x) - f(x^\star) &\leq (\rho - 1)\frac{\mu}{2}\left(\frac{2}{\mu\rho} \|\nabla f(x)\|\right)^{\rho/(\rho-1)} \\ &\leq \left(\frac{2}{\mu}\right)^{1/(\rho-1)} \frac{\rho - 1}{\rho^{\rho/(\rho-1)}} \|\nabla f(x)\|^{\rho/(\rho-1)}. \end{aligned}$$

■

In particular a μ-strongly convex function verifies the Łojasiewicz inequality with $\beta = 2$ and $c = 1/(2\mu)$.

We now prove a similar link between the Tsybakov Noise condition (TNC) and the Łojasiewicz equation.

Proposition 14 *If f is a convex differentiable function locally satisfying the TNC with parameters κ and μ then it satisfies the Łojasiewicz equation with parameters $\kappa/(\kappa-1)$ and $\mu^{-1/(\kappa-1)}$.*

Proof Let $x, y \in \mathbb{R}^d$. Since f is convex we have, noting $x^* = \text{argmin} f$,

$$\begin{aligned} f(y) &\geq f(x) + \langle \nabla f(x), y - x \rangle \\ f(x) - f(x^*) &\leq \langle \nabla f(x), x - x^* \rangle \\ f(x) - f(x^*) &\leq \|\nabla f(x)\| \, \|x - x^*\| . \end{aligned}$$

The TNC gives $f(x) - f(x^*) \geq \mu \|x - x^*\|^\kappa$, which means that $\|x - x^*\| \leq \mu^{-1/\kappa} \left(f(x) - f(x^*)\right)^{1/\kappa}$ and consequently,

$$\begin{aligned} f(x) - f(x^*) &\leq \|\nabla f(x)\| \, \mu^{-1/\kappa} \left(f(x) - f(x^*)\right)^{1/\kappa} \\ \left(f(x) - f(x^*)\right)^{1-1/\kappa} &\leq \mu^{-1/\kappa} \|\nabla f(x)\| \\ \left(f(x) - f(x^*)\right) &\leq \mu^{-1/(\kappa-1)} \|\nabla f(x)\|^{\kappa/(\kappa-1)} . \end{aligned}$$

This concludes the proof. ■

We now show that the two classes of uniformly convex functions and Łojasiewicz functions are distinct by giving examples of functions that verify the Łojasiewicz inequality and that are not uniformly convex.

Example 1 *The function $f : (x, y) \in \mathbb{R}^2 \mapsto (x - y)^2$ verifies the Łojasiewicz inequality but is not uniformly convex on $\mathbb{R}^2$.*

Proof $\nabla f(x, y) = 2(x - y, y - x)^\top$ and $\|\nabla f(x, y)\|^2 = 8(x - y)^2 = 8 f(x, y)$. Consequently, since f is minimal at 0, f verifies the Łojasiewicz inequality for $\beta = 2$ and $c = 1/8$.

Let $a = (0, 0)$ and $b = (1, 1)$. If f is uniformly convex on $\mathbb{R}^2$ with parameters ρ and μ then, for $\alpha = 1/2$,

$$\begin{aligned} f(a/2 + b/2) &\leq f(a)/2 + f(b)/2 - \mu/4(2^{1-\rho}) \|a - b\|^\rho \\ 0 &\leq -\mu/4(2^{1-\rho})\sqrt{2}^\rho . \end{aligned}$$

This is a contradiction since $\mu > 0$ and $\rho \geq 2$. ■

Example 2 *The function $g : (x, y, z) \in \Delta^3 \mapsto (x - 1)^2 + 2(1 - y) + 2(1 - z)$ is not uniformly convex on the simplex Δ^3 but verifies the Łojasiewicz inequality.*

Proof g is constant on the set $\{x = 0\}$ (since $y + z = 1$). And therefore g is not uniformly convex (take two distinct points in $\{x = 0\}$).

We have $\nabla g(x, y, z) = (2x - 2, -2, -2)^\top$ and $\|\nabla g(x, y, z)\|^2 = 4((x - 1)^2 + 2) \geq 8$. Since $y+z = 1-x$ on Δ^3, we have $g(x, y, z) = (x-1)^2+4-2(1-x) = x^2+3$. Consequently $\min g = 3$. Hence $g(x, y, z) - \min g = x^2 \leq 1 \leq \|\nabla g(x, y, z)\|^2$ and g verifies the Łojasiewicz inequality on Δ^3. ■

We conclude this section by giving additional examples of functions verifying the Łojasiewicz inequality.

Example 3 *If $h : x \in \mathbb{R}^K \mapsto \|x - x^\star\|^\alpha$ with $\alpha \geq 1$. Then h verifies the Łojasiewicz inequality with respect to the parameters $\beta = \alpha/(\alpha - 1)$ and $c = \sqrt{K}$.*

The last example is stated in the concave case because it is an important case of application of our initial problem.

Example 4 *Let $f_1, \ldots, f_K$ be such that $f_k(x) = -a_k x^2 + b_k x$ with $b_k \geq 2a_k \geq 0$. Then $F = \sum_k f_k(x_k)$ satisfies the Łojasiewicz inequality with $\beta = 2$ if at least one a_k is positive. Otherwise, the inequality is satisfied on Δ^K for any $\beta \geq 1$ (with a different constant for each β).*

Proof Indeed, let $x \in \Delta^K$. If there exists at least one positive a_k, then F is quadratic, so if we denote by $x^\star$ its maximum and H its Hessian (it is the diagonal matrix with $-a_k$ on coordinate k), we have

$$F(x) - F(x^\star) = (x - x^\star)^\top H(x - x^\star) \text{ and } \nabla F(x) = 2H(x - x^\star).$$

Hence F satisfies the Łojasiewicz conditions with $\beta = 2$ and $c = 1/(4 \min_k a_k)$. If all f_k are linear, then $F(x^*) - F(x) \leq \max_j b_j - \min_j b_j$ and $\|\nabla F(x)\| = \|b\|$. Given any $\beta \geq 1$, it holds that

$$F(x^*) - F(x) \leq c_\beta \|\nabla F(x)\|^\beta = c_\beta \|b\|^\beta \quad \text{with } c_\beta = (\max_j b_j - \min_j b_j)/\|b\|^\beta .$$

■

Appendix B. Additional results on the complexity class

In this appendix we want to give more precisions on the class of functions we are considering. We give more intuition and we prove some results on examples of classes satisfying our assumption. Finally we will state some properties of the functions of this tree.

B.1. Motivations and examples of sets of functions $\mathcal{F}$ satisfying inductively the Łojasiewicz inequality

First, we recall the definition of the local TNC inequality, around the minimum x^* of a function f with vanishing gradient. More precisely, f satisfies **locally** the TNC if

$$\forall x \in \mathcal{X}, \quad f(x) - \min_{x^* \in \mathcal{X}} f(x^*) \geq \mu \|x - x^*\|^\kappa,$$

where in the above the x^* on the r.h.s. is the minimizer of f the closer to x (in case where f has non-unique minimizer).

Uniform convexity, TNC and Łojasiewicz inequality are connected since it is well known that if a function f is uniformly convex, it satisfies both the local TNC and the Łojasiewicz inequality. Those two concepts are actually equivalent for convex mappings.

The most precise complexity parameter is therefore induced by the Łojasiewicz inequality; however, the mappings f_k considered are increasing on $[0,1]$ hence $\nabla f(x^*)$ might be non-zero and the concept of Łojasiewicz inequality is not appropriate; this is the reason why we need to define the concept of funtions that satisfy *pair-wisely* the Łojasiewicz inequality (see Subsection 2.2).

We provide now examples of functions that satisfy inductively the Łojasiewicz inequality. In particular, a set of functions of cardinality 2 satisfies inductively the Łojasiewicz inequality if and only if these functions satisfy it pair-wisely. Another crucial property of our construction is that if f_{left} and f_{right} are concave, non-decreasing and zero at 0, then these three properties also hold for their parent $x \mapsto \max_{z \leq x} f_{\text{left}}(z) + f_{\text{right}}(x-z)$. As a consequence, if these three properties hold at the leaves, they will hold at all nodes of the tree. See Proposition 15 for similar alternative statements.

Proposition 15 *Assume that $\mathcal{F} = \{f_1, \ldots, f_K\}$ is finite then $\mathcal{F}$ satisfies inductively the Łojasiewicz inequality with respect to some $\beta_{\mathcal{F}} \in [1, +\infty)$. Moreover,*

1. *if f_k are all concave, non-decreasing and $f_k(0) = 0$, then all functions created inductively in the tree satisfy the same assumption.*
2. *If f_k are all ρ-uniformly concave, then so are all the functions created and $\mathcal{F}$ satisfies inductively the Łojasiewicz inequality for $\beta_{\mathcal{F}} \geq \frac{\rho}{\rho-1}$.*
3. *If f_k satisfies the global κ-TNC, then so are all the functions created and $\mathcal{F}$ satisfies inductively the Łojasiewicz inequality for $\beta_{\mathcal{F}} \geq \frac{\kappa}{\kappa-1}$.*
4. *If f_k satisfies the global β-Łojasiewicz inequality, then so are all the functions created and $\mathcal{F}$ satisfies inductively the Łojasiewicz inequality for $\beta_{\mathcal{F}} \geq \beta$.*
5. *If f_k are concave, then $\mathcal{F}$ satisfies inductively the Łojasiewicz inequality w.r.t. $\beta_{\mathcal{F}} = 1$.*
6. *If f_k are linear then $\mathcal{F}$ satisfies inductively the Łojasiewicz inequality w.r.t. any $\beta_{\mathcal{F}} \geq 1$.*
7. *More specifically, if $\mathcal{F}$ is a finite subset of the following class of functions*

$$\mathcal{C}_\alpha := \left\{ x \mapsto \theta(\gamma - x)^\alpha - \theta\gamma^\alpha \,;\, \theta \in \mathbb{R}_-, \gamma \geq 1 \right\}, \quad \text{if } \alpha > 1$$

then $\mathcal{F}$ satisfies inductively the Łojasiewicz inequality with respect to $\beta = \frac{\alpha}{\alpha-1}$.

Proof

1. We just need to prove that the mapping $x \mapsto H(x) = \max_{z \leq x} f_1(z) + f_2(x-z) = \max_{z \leq x} G(z;x)$ satisfies the same assumption as f_1 and f_2, the main question being concavity. Given $x_1, x_2, \lambda \in [0,1]$, let us denote by z_1 the point where $G(\cdot\,; x_1)$ attains its maximum (and similarly z_2 where $G(\cdot\,; x_2)$ attains its maximum). Then the following holds

$$\begin{aligned} H(\lambda x_1 + (1-\lambda)x_2) &\geq f_1(\lambda z_1 + (1-\lambda)z_2) + f_1(\lambda x_1 + (1-\lambda)x_2 - \lambda z_1 - (1-\lambda)z_2) \\ &\geq \lambda f_1(z_1) + (1-\lambda) f_1(z_2) + \lambda f_2(x_1 - z_1) + (1-\lambda) f_2(x_2 - z_2) \\ &= \lambda H(x_1) + (1-\lambda) H(x_2) \end{aligned}$$

   so that concavity is ensured. The fact that $H(0) = 0$ and $H(\cdot)$ is non-decreasing are trivial.

2. Let us prove that the mapping $(x \mapsto H(x) = \max 0 \leq z \leq x f_1(z) + f_2(x-z))$ is also ρ-uniformly concave.

   Let $\alpha \in (0,1)$. Let $(x,y) \in \mathbb{R}^2$. Let us denote by z_x the point in $(0,x)$ such that $H(x) = f_1(z_x) + f_2(x-z_x)$ and by z_y the point in $(0,y)$ such that $H(y) = f_1(z_y) + f_2(y-z_y)$. We have

$$\begin{aligned}
\alpha H(x) + (1-\alpha)H(y) &= \alpha f_1(z_x) + \alpha f_2(x-z_x) + (1-\alpha) f_1(z_y) + (1-\alpha) f_2(y-z_y) \\
&\leq f_1(\alpha z_x + (1-\alpha) z_y) - \frac{\mu}{2}\alpha(1-\alpha)\left(\alpha^{\rho-1} + (1-\alpha)^{\rho-1}\right)\|z_x - z_y\|^\rho \\
&+ f_2(\alpha(x-z_x) + (1-\alpha)(y-z_y)) \\
&- \frac{\mu}{2}\alpha(1-\alpha)\left(\alpha^{\rho-1} + (1-\alpha)^{\rho-1}\right)\|x - z_x - y + z_y\|^\rho \\
&\leq H(\alpha x + (1-\alpha) y) - \frac{\mu}{2}\alpha(1-\alpha)(\|x-y\|/2)^\rho
\end{aligned}$$

   where we used the fact that f_1 and f_2 are ρ-uniformly concave, and the definition of $H(\alpha x + (1-\alpha)y)$, and that $a^\rho + b^\rho \geq ((a+b)/2)^\rho$, for $a, b \geq 0$.

   This proves that H is $(\rho, \mu/2^\rho)$-uniformly convex. Finally Proposition 13 shows that $\mathcal{F}$ satisfies inductively the Łojasiewicz inequality for $\beta_F \geq \rho/(\rho-1)$.

3. Let us use the same notations as in the previous proof. We want to show that H satisfies the global TNC equation.

   Let us suppose that $x \geq y$. We will show first that $z_x \geq z_y$. Let us consider the functions $G_x : z \mapsto f_1(z) + f_2(x-z)$ and $G_y : z \mapsto f_1(z) + f_2(y-z)$.

   - If $z_y = 0$, $z_x \geq z_y$.
   - If $z_y = y$, then $\nabla G_y(y) = \nabla f_1(y) - \nabla f_2(0) \geq 0$. Then $\nabla G_x(y) = \nabla f_1(y) - \nabla f_2(x-y) \geq \nabla f_1(y) - \nabla f_2(0) \geq 0$ since $-\nabla f_2$ is non-decreasing by concavity of f_2. Consequently the maximum of G_x is reached for $z \geq y$ and $z_x \geq z_y$.
   - If $z_y \in (0,y)$. Then $\nabla G_x(z_y) = \nabla f_1(z_y) - \nabla f_2(x-z_y) \geq \nabla f_1(z_y) - \nabla f_2(y-z_y)$. Consequently $\nabla G_x(z_y) \geq \nabla G_y(z_y)$ and $\nabla G_y(z_y) = 0$. Therefore $z_x \geq z_y$.

   We use the exact same proof to show that $x - z_x \geq y - z_y$ (by inverting the roles of f_1 and f_2).

   Using the global κ-TNC for f_1 we get, since it is non-decreasing,

$$f_1(z_x) - f_1(z_y) \geq \mu \|z_x - z_y\|^\kappa$$

   and similarly for f_2:

$$f_2(x-z_x) - f_2(y-z_y) \geq \mu \|x - z_x - y + z_y\|^\kappa .$$

   Summing these inequalities gives

$$H(x) - H(y) \geq \mu/2^\kappa \|x-y\|^\kappa .$$

   This shows that H satisfies the global TNC equation for parameters κ and $\mu/2^\kappa$.

   Proposition 14 finally shows that $\mathcal{F}$ inductively satisfies the Łojasiewicz equation for parameter $\beta_F \geq \kappa/(\kappa-1)$.

4. We still use the same notations as before. We want to show that H satisfies the global Łojasiewicz equation.

$$\begin{aligned}|H(x)-H(y)| &= |f_1(z_x)+f_2(x-z_x)-f_1(z_y)-f_2(y-z_y)| \\ &\leq |f_1(z_x)-f_1(z_y)|+|f_2(x-z_x)-f_2(y-z_y)| \\ &\leq \mu\,\|\nabla f_1(z_x)-\nabla f_1(z_y)\|^\beta+\mu\,\|\nabla f_2(x-z_x)-\nabla f_2(y-z_y)\|^\beta\end{aligned}$$

   In the case where $z_x \notin \{0,x\}$, we have $\nabla H(x)=\nabla f_1(z_x)=\nabla f_2(x-z_x)$. If $z_x=0$, $\nabla H(x)=\nabla f_2(x)>\nabla f_1(0)$ and if $z_x=x$, $\nabla H(x)=\nabla f_1(x)>\nabla f_2(0)$. Let us suppose (without loss of generality) that $x\geq y$.

   - $z_x=0$, then $z_y=0$ (cf previous item) and $\nabla f_2(x-z_x)=\nabla H(x)$ and $\nabla f_2(y-z_y)=\nabla H(y)$ and consequently $|H(x)-H(y)|\leq \mu\,\|\nabla H(x)-\nabla H(y)\|^\beta$.
   - $z_x>0$ so that $\nabla H(x)=\nabla f_1(z_x)$ and $\nabla H(y)\geq \nabla f_1(z_y)$, meaning that $\|\nabla f_1(z_x)-\nabla f_1(z_y)\|\leq \|\nabla H(y)-\nabla H(x)\|$, and a similar analysis shows that $\|\nabla f_1(x-z_x)-\nabla f_1(y-z_y)\|\leq \|\nabla H(y)-\nabla H(x)\|$.

   This means finally that H satisfies the Łojasiewicz equation with parameters β and 2μ.

5. This point is actually a direct consequence of the following Lemma 29.

6. If f_1 and f_2 are linear, then $x\mapsto \max_z \leq x f_1(z)+f_2(x-z)$ is either equal to f_1 or to f_2 (depending on which one is the biggest). Hence it is linear.

7. Assume that $f_i=\theta_i(\gamma_i-x)^\alpha-\theta_i\gamma_i^\alpha$ for some parameter $\gamma_i>1$ and $\theta_i<0$. Then easy computations show that H is equal to either f_1 or f_2 on a small interval near 0 (depending on the size of $\nabla f_i(0)$) and then $H(x)=\theta_0(\gamma_0-x)^\alpha-c_0$ for some parameters $\theta_0<0$ and $\gamma_0>1$. As a consequence, H is defined piecewisely by functions in $\mathcal{C}_\alpha$, a property that will propagate in the binary tree used in the definition of inductive satisfiability of Łojasiewicz inequality.

   The fact that those functions satisfies the Łojasiewicz inequality with respect to $\beta=\frac{\alpha}{\alpha-1}$ has already been proved in Example 3.

■

B.2. Some properties of the functions of the tree

We present now some properties of the functions defined in the labeled tree constructed in the previous section. We begin by a technical and useful lemma.

Lemma 16 *Let f and g be two differentiable concave functions on $[0,1]$. For $x\in[0,1]$ define $\phi_x: z\in[0,x]\mapsto f(z)+g(x-z)$. And $z_x \doteq \text{argmax}_{z\in[0,x]}\phi_x(z)$. We have the following results:*

- *ϕ_x is concave;*
- *$\forall\ 0\leq x\leq y\leq 1,\ z_x\leq z_y$ and $x-z_x\leq y-z_y$. In particular the function $x\mapsto z_x$ is 1-Lipschitz continuous.*

Proof *The fact that ϕ_x is concave is immediate since f and g are concave functions.*

If $0 \leq x \leq y \leq 1$, we have $g'(y - z_x) \leq g'(x - z_x)$ since $y - z_x \geq x - z_x$ and g' is non-increasing (because g is concave). Consequently, $\phi'_y(z_x) = f'(z_x) - g'(y - z_x) \geq \phi'_x(z_x)$. If $z_x = 0$, $z_y \geq z_x$ is immediate. Otherwise, $z_x > 0$ and $\phi'(z_x) \geq 0$. This shows that $\phi'_y(z_x) \geq 0$ and consequently, that the maximum z_y of the concave function ϕ_y is reached after z_x. And $z_y \geq z_x$.

The last inequality is obtained in a symmetrical manner by considering the function $\psi_x : z \in [0, x] \mapsto f(x - z) + g(x)$ whose maximum is reached at $z = x - z_x$. This gives $x - z_x \leq y - z_y$. ■

We now proove two simples lemmas.

Lemma 17 *If f and g are two concave L-Lipschitz continous and differentiable functions, then $H : x \mapsto \max_{z \in [0,x]} f(z) + g(x - z)$ is L-Lipschitz continuous.*

Proof With the notations of the previous lemma, we have $H(x) = \phi_x(z_x)$ for all $x \in [0, 1]$.

Let $x, y \in [0, 1]$. Without loss of generality we can suppose that $x \leq y$. We have

$$\begin{aligned} |H(x) - H(y)| &= |f(z_x) + g(x - z_x) - f(z_y) - g(y - z_y)| \\ &\leq L|z_x - z_y| + L|x - z_x - (y - z_y| \\ &\leq L(z_y - z_x) + L(y - z_y - x + z_x) \\ &\leq L|y - x|. \end{aligned}$$

We have used the conclusion of Lemma 16 in the third line. ■

Lemma 18 *If f and g are two concave L'-smooth and differentiable functions, then $H : x \mapsto \max_{z \in [0,x]} f(z) + g(x - z)$ is L'-smooth.*

Proof Let $x, y \in [0, 1]$. Without loss of generality we can suppose that $x \leq y$. We treat the case where $\phi_x \in (0, x)$ and $\phi_y \in (0, y)$. The other (extremal) cases can be treated similarly. The envelop theorem gives that $\nabla H(x) = \nabla f(z_x)$ and $\nabla H(y) = \nabla f(z_y)$. Therefore $|\nabla H(x) - \nabla H(y)| = |\nabla f(z_x) - \nabla f(z_y)| \leq L'|z_x - z_y| \leq L'|x - y|$ with Lemma 16. ■

Proposition 15 and Lemmas 17 and 18 show directly the following proposition:

Proposition 19 *If the functions $f_1, \ldots, f_K$ are concave differentiable L-Lipschitz continuous and L'-smooth then all functions created in the tree are also concave differentiable L-Lipschitz continuous and L'-smooth.*

B.3. Proof of Proposition 2

We begin by proving the following lemma:

Lemma 20 *If f and g are strictly concave real analytic functions then $H : x \mapsto \max_{0 \leq z \leq x} f(z) + g(x - z)$ is also a strictly concave real analytic function.*

Proof The fact that H is strictly concave comes from Proposition 15. Since f and g are real analytic functions we can write

$$f(x) = \sum_{n\geq 0} a_n x^n \quad \text{and} \quad g(x) = \sum_{n\geq 0} b_n x^n.$$

Let us consider the function $\phi_x : z \mapsto f(z) + g(x-z)$ for $z \in [0,x]$. Now, for all $0 \leq z \leq x$, we have

$$\begin{aligned}
\phi_x(z) &= f(z) + g(x-z) \\
&= \sum_{n\geq 0} a_n z^n + \sum_{n\geq 0} b_n (x-z)^n \\
&= \sum_{n\geq 0} a_n z^n + \sum_{n\geq 0} b_n \sum_{k=0}^{n} \binom{n}{k} x^{n-k} (-1)^k z^k \\
&= \sum_{k\geq 0} a_k z^k + \sum_{k\geq 0} \left(\sum_{n\geq k} b_n (-1)^k x^{n-k} \right) z^k \\
&= \sum_{k\geq 0} c_k(x) z^k,
\end{aligned}$$

with $c_k(x) = a_k + \sum_{n\geq k} b_n (-1)^k x^{n-k}$.

Since f and g are concave, ϕ_x is also concave. Let $z_x \doteq \operatorname{argmax}_{z\in[0,x]} \phi_x(z)$. We have $H(x) = \phi_x(z_x)$ If $z_x \in (0,x)$ then $\nabla \phi_x(z_x) = 0$ because ϕ_x is concave. Consequently $\sum_{k\geq 0} c_{k+1}(x)(k+1) z_x^k = 0$.

Let us consider the function $\Psi : (x,z) \mapsto \sum_{k\geq 0} c_{k+1}(x)(k+1) z_x^k = \nabla \phi_x(z)$. Provided that $\nabla_z \Psi(x, z_x)$ is invertible then z_x is unique and is an analytic function of x thanks to the analytic implicit function theorem (Berger, 1977). Since f and g are strictly concave the invertibility condition is satisfied since $\nabla_z \Psi(x, z_x) = f''(z) + g''(x-z)$, and the result is proved. ■

Proof of Proposition 2. Let us show that $\mathcal{F}$ satisfies inductively the Łojasiewicz inequality. Let f and g be two siblings of the tree defined in Appendix B. Inductively applying Lemma 20 shows that $(x \mapsto \max_{0\leq z\leq x} f(z) + g(x-z))$ is a strictly concave real analytic function. Since a real analytic function verifies the Łojasiewicz inequality (Łojasiewicz, 1965), the result is proved. We set β to be the maximum of all Łojasiewicz exponents in the tree. ■

Appendix C. Analysis of the algorithm with $K = 2$ resources

C.1. Proof of Theorem 4

Proof Let $j \in [j_{\max}]$. By concavity of g, we have that $-g(x_j) \leq |g'(x_j)||x^\star - x_j|$. Since g is negative, this means that $|g(x_j)| \leq |g'(x_j)||x^\star - x_j|$.

Since g is of class $\mathcal{C}^2$ and α-strongly concave,

$$\begin{aligned}\langle g'(x_j) - g'(x^\star) | x_j - x^\star \rangle &\leq -\alpha \|x_j - x^\star\|^2 \\ -\alpha \|x_j - x^\star\|^2 \geq \langle g'(x_j) - g'(x^\star) | x_j - x^\star \rangle &\geq -|g'(x_j)| \|x_j - x^\star\| \\ |g'(x_j)| &\geq \alpha \|x_j - x^\star\|.\end{aligned}$$

Then

$$\frac{|g(x_j)|}{g'(x_j)^2} \leq \frac{|g'(x_j)||x^\star - x_j|}{g'(x_j)^2} = \frac{|x^\star - x_j|}{|g'(x_j)|} \leq \frac{1}{\alpha}.$$

Consequently we have

$$R(T) \leq \frac{j_{\max}}{T\alpha}.$$

We have for all $j \in [j_{\max}]$, $N_j = 2\log(2T/\delta)\dfrac{1}{g'(x_j)^2}$. Then

$$\begin{aligned}T &= 8\log(2T/\delta) \sum_{j=1}^{j_{\max}} \frac{1}{g'(x_j)^2} \\ &\geq 8\log(2T/\delta) \sum_{j=1}^{j_{\max}} \frac{1}{L'^2(x_j - x^\star)^2} \\ &\geq 8\log(2T/\delta) \frac{1}{L'^2(x_{j_{\max}} - x^\star)^2} \\ &\geq 8\log(2T/\delta) \frac{4^{j_{\max}}}{L'^2}.\end{aligned}$$

where we used the fact that g' is L'-Lipschitz continuous. Therefore $j_{\max} \leq \log_4\left(\dfrac{TL'^2}{8\log(2T/\delta)}\right) \lesssim \log(T)$. And finally

$$R(T) = \mathcal{O}\left(\frac{1}{\alpha}\frac{\log(T)}{T}\right).$$

■

C.2. Proof of Theorem 5, when $\beta > 2$

Proof Let $x \in [0,1]$. We know that $|g(x)| \leq c|g'(x)|^\beta$.

Then $\dfrac{1}{|g'(x)|^2} \leq \dfrac{c^{2/\beta}}{|g(x)|^{2/\beta}}$, and $\dfrac{|g(x)|}{|g'(x)|^2} \leq c^{2/\beta}|g(x)|^{1-2/\beta}$.

Since g is L-Lipschitz on $[0,1]$, we have $|g(x) - g(x^\star)| \leq L|x - x^\star|$. Since $g(x^\star) = 0$ then $\dfrac{|g(x)|}{|g'(x)|^2} \leq c^{2/\beta}L^{1-2/\beta}|x - x^\star|^{1-2/\beta}$.

For $j \in [j_{\max}]$, $\dfrac{|g(x_j)|}{|g'(x_j)|^2} \leq c^{2/\beta}L^{1-2/\beta}\left(\dfrac{1}{2^{1-2/\beta}}\right)^j$, because $|x^\star - x_j| \leq 2^{-j}$, as a consequence of the binary search. Since $1 - 2/\beta > 0$,

$$\sum_{j=1}^{j_{\max}} \left(\frac{1}{2^{1-2/\beta}}\right)^j < \frac{1}{1 - 2^{2/\beta - 1}}.$$

Finally we have, using that $\delta = 2/T^2$,

$$\begin{aligned} R(T) &= \frac{8}{T}\log(2T/\delta)\sum_{j=1}^{j_{\max}}\frac{|g(x_j)|}{|g'(x_j)|^2} \\ &\leq \frac{24c^{2/\beta}L^{1-2/\beta}}{1-2^{2/\beta-1}}\frac{\log(T)}{T}. \end{aligned}$$

■

C.3. Proof of Theorem 5, when $\beta < 2$

Proof We know that

$$\begin{aligned} R(T) &= \frac{1}{T}\sum_{j=1}^{j_{\max}}|g(x_j)|N_j \\ &= 8\log(2T/\delta)\frac{1}{T}\sum_{j=1}^{j_{\max}}\frac{|g(x_j)|}{h_j^2} \\ &\leq 8\log(2T/\delta)\frac{1}{T}\sum_{j=1}^{j_{\max}}\frac{|g(x_j)|}{g'(x_j)^2}. \end{aligned}$$

where $h_j \geq g_j$ is such that $N_j = \dfrac{8\log(2/\delta)}{h_j^2}$. We note

$$R \doteq \frac{TR(T)}{8\log(2T/\delta)} = \sum_{j=1}^{j_{\max}}\frac{|g(x_j)|}{h_j^2}.$$

By hypothesis, $\forall x \in [0,1],\ |g(x)| \leq c|g'(x)|^\beta$. Moreover Lemma 29 gives $|g(x_j)| \leq |g'(x_j)||x_j - x^\star| \leq |g'(x_j)|2^{-j}$.

If we note $g_j \doteq |g'(x_j)|$ we obtain

$$R \leq \sum_{j=1}^{j_{\max}}\min\left(cg_j^\beta, \frac{g_j}{2^j}\right)\frac{1}{h_j^2}.$$

Let us now note

$$T' \doteq \frac{T}{8\log(2T/\delta)}.$$

We have the constraint

$$T' = \sum_{j=1}^{j_{\max}}\frac{1}{h_j^2}.$$

Our goal is to bound R. In order to do that, one way is to consider the functional

$$\mathcal{F} : (g_1, \ldots, g_{j_{\max}}) \in {\mathbb{R}_+^*}^{j_{\max}} \mapsto \sum_{j=1}^{j_{\max}}\min\left(cg_j^\beta, \frac{g_j}{2^j}\right)/h_j^2$$

and to maximize it under the constraints

$$T' = \sum_{j=1}^{j_{\max}} \frac{1}{h_j^2} \quad \text{and } g_j \leq h_j.$$

Therefore the maximum of the previous problem is smaller than the one of maximizing

$$\hat{\mathcal{F}} : (h_1, \ldots, h_{j_{\max}}) \in {\mathbb{R}_+^*}^{j_{\max}} \mapsto \sum_{j=1}^{j_{\max}} \min\left(c h_j^{\beta-2}, \frac{1}{h_j 2^j} \right)$$

and to maximize it under the constraints

$$T' = \sum_{j=1}^{j_{\max}} \frac{1}{h_j^2}.$$

For the sake of simplicity we identify g_j with h_j. The maximization problem can be done with Karush-Kuhn-Tucker conditions: introducing the Lagrangian

$$\mathcal{L}(g_1, \ldots, g_{j_{\max}}, \lambda) = \mathcal{F}(g_1, \ldots, g_{j_{\max}}) + \lambda \left(T' - \sum_{j=1}^{j_{\max}} \frac{1}{h_j^2} \right)$$

we obtain

$$\frac{\partial \mathcal{L}}{\partial g_j} = \begin{cases} c(\beta-2) g_j^{\beta-3} + \dfrac{2\lambda}{g_j^3}, & \text{if } g_j < \hat{g}_j \\ -\dfrac{1}{2^j g_j} + \dfrac{2\lambda}{g_j^3}, & \text{if } g_j > \hat{g}_j \end{cases}, \text{where } \hat{g}_j = \left(\frac{1}{2^j c} \right)^{1/(\beta-1)}.$$

$\hat{g}_j$ is the point where the two quantities in the $\min$ are equal. And finally

$$\begin{cases} g_j = \left(\dfrac{2\lambda}{c(2-\beta)} \right)^{1/\beta}, & \text{if } g_j < \hat{g}_j \\ g_j = 2\lambda \cdot 2^j, & \text{if } g_j > \hat{g}_j. \end{cases}$$

We note $\mathcal{J}_1 \doteq \{ j \in [j_{\max}], g_j > \hat{g}_j \}$ and $\mathcal{J}_2 \doteq \{ j \in [j_{\max}], g_j < \hat{g}_j \}$. We have

$$\mathcal{F}(g_1, \ldots, g_{j_{\max}}) = \underbrace{\sum_{j \in \mathcal{J}_1} \frac{1}{2^j g_j}}_{\mathcal{F}_1} + \underbrace{\sum_{j \in \mathcal{J}_2} c g_j^{\beta-2}}_{\mathcal{F}_2}.$$

We note as well

$$T_1 \doteq \sum_{j \in \mathcal{J}_1} \frac{1}{g_j^2} \quad \text{and} \quad T_2 \doteq \sum_{j \in \mathcal{J}_2} \frac{1}{g_j^2} \quad \text{such that } T' = T_1 + T_2.$$

on $\mathcal{J}_2$:

Since $g_j < \hat{g}_j$ on $\mathcal{J}_2$, noting $g_2 \doteq \left(\frac{2\lambda}{c(2-\beta)}\right)^{1/\beta} = g_j$,

$$T_2 = \sum_{j\in\mathcal{J}_2} \frac{1}{g_j^2} = |\mathcal{J}_2|\frac{1}{g_2^2} > |\mathcal{J}_2|\frac{1}{\hat{g}_j^2} \text{ for all } j \in \mathcal{J}_2$$

In particular,

$$T' \geq T_2 > |\mathcal{J}_2| \left(\frac{1}{c^2 4^{j_{2,\max}}}\right)^{-1/(\beta-1)} \geq |\mathcal{J}_2| \left(c^2 4^{|\mathcal{J}_2|}\right)^{1/(\beta-1)} \geq \left(4^{|\mathcal{J}_2|}\right)^{1/(\beta-1)}$$

because c can be chosen greater than 1. This gives $|\mathcal{J}_2| \leq \frac{\beta-1}{\log(4)}\log(T)$.

And we know that

$$T_2 = \sum_{j\in\mathcal{J}_2} \frac{1}{g_j^2} = |\mathcal{J}_2| \left(\frac{2\lambda}{c(2-\beta)}\right)^{-2/\beta}.$$

This gives

$$\frac{2\lambda}{c(2-\beta)} = \left(\frac{T_2}{|\mathcal{J}_2|}\right)^{-\beta/2}.$$

We can now compute the cost of $\mathcal{J}_2$:

$$\begin{aligned}
\mathcal{F}_2 &= \sum_{j\in\mathcal{J}_2} c g_j^{\beta-2} \\
&= |\mathcal{J}_2| c \left(\frac{2\lambda}{c(2-\beta)}\right)^{(\beta-2)/\beta} \\
&= |\mathcal{J}_2| c \left(\frac{T_2}{|\mathcal{J}_2|}\right)^{1-\beta/2} \\
&= c T_2^{1-\beta/2} |\mathcal{J}_2|^{\beta/2} \\
&\leq c T_2^{1-\beta/2} \left(\frac{\beta-1}{\log(4)}\log(T')\right)^{\beta/2} \\
&\lesssim c T' \left(\frac{\log(T')}{T'}\right)^{\beta/2}.
\end{aligned}$$

on $\mathcal{J}_1$:

We know that $\forall j \in \mathcal{J}_1,\ g_j = 2\lambda\, 2^j$. This gives

$$\begin{aligned}
T_1 &= \sum_{j\in\mathcal{J}_1} \frac{1}{g_j^2} = \frac{1}{4\lambda^2} \sum_{j\in\mathcal{J}_1} \frac{1}{4^j} \\
2\lambda &= \sqrt{\frac{\sum_{j\in\mathcal{J}_1} 4^{-j}}{T_1}} \\
2\lambda &\leq \sqrt{\frac{4\cdot 4^{-j_{1,\min}}}{3T_1}}.
\end{aligned}$$

Since $j \in \mathcal{J}_1$, we know that $g_j \geq \hat{g}_j$ and $2\lambda\, 2^j \geq \left(\frac{1}{2^j c}\right)^{1/(\beta-1)}$, and $2\lambda \geq c^{-1/(\beta-1)}(2^j)^{-\beta/(\beta-1)}$. With $j = j_{1,\min}$ we obtain

$$\begin{aligned} c^{-1/(\beta-1)}(2^{j_{1,\min}})^{-\beta/(\beta-1)} &\leq \sqrt{\frac{4 \cdot 4^{-j_{1,\min}}}{3T_1}} \\ \frac{\sqrt{3}}{2}\left(2^{j_{1,\min}}\right)^{-1/(\beta-1)} c^{-1/(\beta-1)} &\leq \frac{1}{\sqrt{T_1}} \\ c^{-2}4^{-j_{1,\min}} &\lesssim T_1^{1-\beta}. \end{aligned}$$

And we have

$$\begin{aligned} \mathcal{F}_1 = \sum_{j\in\mathcal{J}_1} \frac{1}{2^j\ 2\lambda\ 2^j} &= \frac{1}{2\lambda}\sum_{j\in\mathcal{J}_1}\frac{1}{4^j} = 2\lambda T_1 \\ &\lesssim \sqrt{T_1}2^{-j_{1,\min}} \\ &\lesssim cT_1^{1-\beta/2} \lesssim cT'^{1-\beta/2}. \end{aligned}$$

Finally we have shown that $R \lesssim cT'\left(\frac{\log(T')}{T'}\right)^{\beta/2}$ and consequently

$$\begin{aligned} \frac{TR(T)}{8\log(2T/\delta)} &\lesssim c\frac{T}{8\log(2T/\delta)}\left(\frac{\log(T')}{T'}\right)^{\beta/2} \\ R(T) &\lesssim c\left(8\log(2T/\delta)\right)^{\beta/2}\left(\frac{\log(T)}{T}\right)^{\beta/2}. \end{aligned}$$

And using the fact that $\beta < 2$ and $\delta = 2/T^2$, we have

$$R(T) \lesssim c\left(\frac{\log(T)^2}{T}\right)^{\beta/2}.$$

■

C.4. Proof of Theorem 5, when $\beta = 2$

Proof As in the previous proof, we want to bound

$$R = \sum_{j=1}^{j_{\max}} \frac{|g(x_j)|}{g'(x_j)^2} \leq \sum_{j=1}^{j_{\max}} \min\left(c, \frac{1}{g_j 2^j}\right).$$

Let us note $\hat{g}_j \doteq \frac{1}{c2^j}$, we have to distinguish two cases:

$$\begin{cases} \text{if } g_j > \hat{g}_j, \quad \text{then } \min\left(c, \frac{1}{g_j 2^j}\right) = \frac{1}{2^j g_j} \\ \text{if } g_j < \hat{g}_j, \quad \text{then } \min\left(c, \frac{1}{g_j 2^j}\right) = c. \end{cases}$$

We note $\mathcal{J}_1 \doteq \{j \in [j_{\max}], g_j > \hat{g}_j\}$ and $\mathcal{J}_2 \doteq \{j \in [j_{\max}], g_j < \hat{g}_j\}$.

We have

$$R \leq \underbrace{\sum_{j\in\mathcal{J}_1} \frac{1}{2^j g_j}}_{R_1} + \underbrace{\sum_{j\in\mathcal{J}_2} c}_{R_2}.$$

We note as well

$$T_1 \doteq \sum_{j\in\mathcal{J}_1} \frac{1}{g_j^2} \quad \text{and} \quad T_2 \doteq \sum_{j\in\mathcal{J}_2} \frac{1}{g_j^2} \quad \text{such that } T' = T_1 + T_2.$$

on $\mathcal{J}_2$:

$$T_2 = \sum_{j\in\mathcal{J}_2} \frac{1}{g_j^2} > \sum_{j\in\mathcal{J}_2} \frac{1}{\hat{g}_j^2} \geq \sum_{j\in\mathcal{J}_2} c^2 4^j \geq 4^{j_{2,\max}}.$$

Which gives $j_{2,\max} \leq \log(T)$. Finally,

$$R_2 = \sum_{j\in\mathcal{J}_2} c \leq c j_{2,\max} \leq c \log(T).$$

on $\mathcal{J}_1$:

We want to maximize $R_1 = \sum_{j\in\mathcal{J}_1} \frac{1}{2^j g_j}$ under the constraint $T_1 = \sum_{j\in\mathcal{J}_1} \frac{1}{g_j^2}$.

Karush-Kuhn-Tucker conditions give the existence of $\lambda > 0$ such that for all $j \in \mathcal{J}_1$, $g_j = 2\lambda \cdot 2^j$. As in the previous proof this shows that $R_1 = 2\lambda T_1$. We can show as well that, if $j \in \mathcal{J}_1$,

$$2\lambda \leq \frac{2}{\sqrt{3}} \frac{2^{-j_{1,\min}}}{\sqrt{T_1}}.$$

And since $j \in \mathcal{J}_1$, $g_j > \frac{1}{c2^j}$ and then $2\lambda\, 2^j > \frac{1}{c2^j}$ which means

$$2\lambda > \frac{1}{c4^{j_{1,\min}}}.$$

Putting these inequalities together gives

$$\sqrt{T_1} \leq \frac{2c}{\sqrt{3}} 2^{j_{1,\min}}.$$

Finally,

$$R_1 = 2\lambda T_1 \leq \frac{2}{\sqrt{3}} \frac{2^{-j_{1,\min}}}{\sqrt{T_1}} T_1 \leq \frac{4c}{3}.$$

This shows that

$$R(T) \lesssim c \log(2T/\delta) \frac{\log(T)}{T} \lesssim c \frac{\log(T)^2}{T}.$$

■

C.5. Proof of Theorem 6

Proof The proof is very similar to the one of Shamir (2013) (see also (Bach and Perchet, 2016)) so we only provide the main different ingredients.

Given T and β, we are going to construct 2 pairs of functions f_1, f_2 and $\widetilde{f}_1, \widetilde{f}_2$ such that

$$\|f_i - \widetilde{f}_i\|_\infty \leq \frac{c_\beta}{\sqrt{T}} \quad \text{and} \quad \|\nabla f_i - \nabla \widetilde{f}_i\|_\infty \leq \frac{c_\beta}{\sqrt{T}}\,.$$

As a consequence, using only T samples[4], it is impossible to distinguish between the pair f_1, f_2 and the pair $\widetilde{f}_1, \widetilde{f}_2$. And the regret incurred by any algorithm is then lower-bounded (up to some constant) by

$$\min_x \max\{g^\star - g(x)\,;\, \widetilde{g}^\star - \widetilde{g}(x)\}$$

where we have defined $g(x) = f_1(x) + f_2(1-x)$ and $g^\star = \max_x g(x)$ and similarly for $\widetilde{g}$.

To define all those functions, we first introduce g and $\widetilde{g}$ defined as follows, where γ is a parameter to be fixed later.

$$g : x \mapsto \begin{cases} -x^{\beta/(\beta-1)} & \text{if } x \leq \gamma \\ -\dfrac{\beta}{\beta-1}\gamma^{1/(\beta-1)}x + \dfrac{1}{\beta-1}\gamma^{\beta/(\beta-1)} & \text{otherwise} \end{cases}$$

and

$$\widetilde{g} : x \mapsto \begin{cases} -|x-\gamma|^{-\beta/(\beta-1)} & \text{if } x \leq 2\gamma \\ -\dfrac{\beta}{\beta-1}\gamma^{1/(\beta-1)}x + \dfrac{\beta+1}{\beta-1}\gamma^{\beta/(\beta-1)} & \text{otherwise.} \end{cases}$$

The functions have the form of Proposition 15 near 0 and then are linear with the same slope. Proposition 15 ensures that g_1 and g_2 verify the Łojasiewicz inequality for the parameter β. The functions g_1 and g_2 are concave non-positive functions, reaching their respective maxima at 0 and γ.

We also introduce a third function h defined by

$$h : x \mapsto \begin{cases} (\gamma - x)^{\beta/(\beta-1)} - x^{\beta/(\beta-1)} & \text{if } \frac{\gamma}{2} \leq x \leq \gamma \\ 2\dfrac{\beta}{\beta-1}(\frac{\gamma}{2})^{1/(\beta-1)}(\frac{\gamma}{2} - x) & \text{if } x \leq \frac{\gamma}{2} \\ -\dfrac{\beta}{\beta-1}\gamma^{1/(\beta-1)}x + \dfrac{1}{\beta-1}\gamma^{\beta/(\beta-1)} & \text{if } x \geq \gamma \end{cases}$$

The functions f_i and $\widetilde{f}_i$ are then defined as

$$f_1(x) = 0 \quad \text{and} \quad \widetilde{f}_1(x) = \widetilde{g}(x) - g(x) + h(x) - \widetilde{g}(0) - g(0) + h(0)$$
$$f_2(x) = g(1-x) - g(1) \quad \text{and} \quad \widetilde{f}_2(x) = g(1-x) - h(1-x) - g(1) + h(1)$$

It immediately follows that $f_1(x) + f_2(1-x)$ is equal to $g(x)$ and similarly $\widetilde{f}_1(x) + \widetilde{f}_2(1-x)$ is equal to $\widetilde{g}(x)$ (both up to some additive constant).

4. Formally, we just need to control the ℓ_∞ distance between the gradients, as we assume that the feedbacks of the decision maker are noisy gradients. But we could have assumed that he also observes noisy evaluations of $f_1(x_1)$ and $f_2(x_2)$. This is why we also want to control the ℓ_∞ distance between the functions f_i and $\widetilde{f}_i$.

We observe that for all $x \in [0,1]$:

$$\nabla g(x) = \begin{cases} -\frac{\beta}{\beta-1}x^{1/(\beta-1)} & \text{if } x \leq \gamma \\ -\frac{\beta}{\beta-1}\gamma^{1/(\beta-1)} & \text{otherwise} \end{cases}$$

and

$$\nabla \widetilde{g}(x) = \begin{cases} -\frac{\beta}{\beta-1}\text{sign}(x-\gamma)|x-\gamma|^{1/(\beta-1)} & \text{if } x \leq 2\gamma \\ -\frac{\beta}{\beta-1}\gamma^{1/(\beta-1)} & \text{otherwise} \end{cases}$$

Similarly, we can easily compute the gradient of h:

$$\nabla h(x) = \begin{cases} -\frac{\beta}{\beta-1}\Big((\gamma-x)^{1/(\beta-1)} + x^{1/(\beta-1)}\Big) & \text{if } \frac{\gamma}{2} \leq x \leq \gamma \\ -2\frac{\beta}{\beta-1}(\frac{\gamma}{2})^{1/(\beta-1)} & \text{if } x \leq \frac{\gamma}{2} \\ -\frac{\beta}{\beta-1}\gamma^{1/(\beta-1)} & \text{if } x \geq \gamma \end{cases}$$

We want to bound $\|\nabla g - \nabla \widetilde{g}\|_\infty$ as it is clear that $\|\nabla h\|_\infty \leq \frac{\beta}{\beta-1}\gamma^{1/(\beta-1)}$.

- For $x \leq \gamma$,

$$\begin{aligned} |\nabla g(x) - \nabla \widetilde{g}(x)| &= \frac{\beta}{\beta-1}\Big|-x^{1/(\beta-1)} - (\gamma-x)^{1/(\beta-1)}\Big| \\ &= \frac{\beta}{\beta-1}\Big|x^{1/(\beta-1)} + (\gamma-x)^{1/(\beta-1)}\Big| \\ &\leq \frac{\beta}{\beta-1}\left(x^{1/(\beta-1)} + (\gamma-x)^{1/(\beta-1)}\right) \\ &\leq 2\frac{\beta}{\beta-1}\gamma^{1/(\beta-1)}. \end{aligned}$$

- For $\gamma \leq x \leq 2\gamma$,

$$\begin{aligned} |\nabla g(x) - \nabla \widetilde{g}(x)| &= \frac{\beta}{\beta-1}\Big|(x-\gamma)^{1/(\beta-1)} - x^{1/(\beta-1)}\Big| \\ &\leq \frac{\beta}{\beta-1}\Big|(x-\gamma)^{1/(\beta-1)}\Big| + \Big|x^{1/(\beta-1)}\Big| \\ &\leq (1+2^{1/(\beta-1)})\frac{\beta}{(\beta-1)}\gamma^{1/(\beta-1)} \end{aligned}$$

- For $x \geq 2\gamma$, $|\nabla g(x) - \nabla \widetilde{g}(x)| = 0$.

Finally we also have that $\|\nabla g - \nabla \widetilde{g}\|_\infty \lesssim \gamma^{1/(\beta-1)}$, where the notation $\lesssim$ hides a multiplicative constant factor.

Combining the control on $\|\nabla g - \nabla \widetilde{g}\|_\infty$ and $\|\nabla h\|_\infty$, we finally get that

$$\left\|\nabla f_1 - \nabla \widetilde{f}_1\right\|_\infty \lesssim \gamma^{1/(\beta-1)} \quad \text{and} \quad \left\|\nabla f_2 - \nabla \widetilde{f}_2\right\|_\infty \lesssim \gamma^{1/(\beta-1)}.$$

As a consequence, the specific choice of $\gamma = T^{(1-\beta)/2}$ ensures that $\gamma^{1/(\beta-1)} \leq 1/\sqrt{T}$ and thus the mappings f_i are indistinguishable from $\widetilde{f}_i$,

Finally, we get

$$R(T) \geq T \min_x \max(|g(x)|, |\widetilde{g}(x)|) \geq T g(\gamma/2) \gtrsim \gamma^{\beta/(\beta-1)} \gtrsim T^{-\beta/2}.$$

■

Appendix D. Analysis of the algorithm with $K > 2$ resources

In this section we present a detailed description of the maximization algorithm introduced in Section 4 as well as a thorough analysis of its complexity.

D.1. Detailed description of the algorithm

The goal of the algorithm is to maximize the following function on the simplex Δ^K:

$$F(x) = \sum_{k=1}^{K} f_k(x_k) \quad \text{with } x = (x_1, \dots, x_K) \in \Delta^K.$$

As mentioned in the main text, the idea is to use a divide and conquer strategy in order to be able to use the procedure of $K = 2$ resources explained in Section 3. The overall idea is to separate arms recursively into two bundles, creating the aforementioned tree whose root is F and whose leaves are the f_k. We explain in this section the algorithm with more details, introducing the relevant definitions and notations for the proof.

We will denote by $F_j^{(i)}$ the function created at the nodes of depth i, with j an increasing index from the left to the right of the tree; in particular $F_1^{(0)} = F = \sum_{k=1}^K f_k(x_k)$. This is the function we want to maximize.

Definition 21 *Starting from $F_1^{(0)} = F = \sum_{k=1}^K f_k(x_k)$, the functions $F_j^{(i)}$ are constructed inductively as follows. If $F_j^{(i)}(x) = \sum_{k=k_1}^{k_2} f_k(x_k)$ is not a leaf (i.e., $k_1 < k_2$) we define*

$$F_{2j-1}^{(i+1)}(x) = \sum_{k=k_1}^{\lfloor (k_1+k_2)/2 \rfloor} f_k(x_k) \quad \textit{and} \quad F_{2j}^{(i+1)}(x) = \sum_{k=\lfloor (k_1+k_2)/2 \rfloor + 1}^{k_2} f_k(x_k).$$

The optimization of $F_j^{(i)}$ can be done recursively since

$$\max_{\|x\|_1 = z_n} F_j^{(i)}(x) = \max_{z_{n+1} \in [0, z_n]} \left(\max_{\|x\|_1 = z_{n+1}} F_{2j-1}^{(i+1)}(x) + \max_{\|x\|_1 = z_n - z_{n+1}} F_{2j}^{(i+1)}(x) \right).$$

The recursion ends at nodes that are parents of leaves, where the optimization problem is reduced to the case of $K = 2$ resources studied in the previous section.

For the sake of notations, we introduce the following functions.

Definition 22 *For every i and j in the constructed binary tree of functions,*

$$H_j^{(i)}(z) \triangleq \max_{\|x\|_1=z} F_j^{(i)}(x) \quad \textit{and} \quad G_j^{(i)}(z;y) \triangleq H_{2j-1}^{(i+1)}(z) + H_{2j}^{(i+1)}(y-z).$$

With these notations, it holds that for all $z_n \in [0,1]$,

$$H_j^{(i)}(z_n) = \max_{z_{n+1}\in[0,z_n]} G_j^{(i)}(z_{n+1};z_n) = \max_{z_{n+1}\in[0,z_n]} H_{2j-1}^{(i+1)}(z_{n+1}) + H_{2j}^{(i+1)}(z_n - z_{n+1}).$$

The computation of $H_j^{(i)}(z_n)$ is made with similar techniques than in the case $K=2$ of Section 3. The idea will be to imbricate several binary searches to get estimates of the functions $\nabla H_j^{(i)}$.

More precisely, to maximize the function $\left(u \mapsto G_j^{(i)}(u;z_n)\right)$ a binary search is run over $[0,z_n]$, starting at $u_1 = z_n/2$:

Definition 23 *We note $\mathcal{D}_j^{(i)}(v)$ the binary search run to maximize $\left(w \mapsto G_j^{(i)}(w;v)\right)$. We define ${z^\star}_j^{(i)}(v)$ as argmax $G_j^{(i)}(\cdot\,;v)$ and we also call $T_j^{(i)}(v)$ the total number of queries used by $\mathcal{D}_j^{(i)}(v)$.*

Inductively, the binary search $\mathcal{D}_j^{(i)}(v)$ searches on the left or on the right of u_m, depending on the sign of $\nabla G_j^{(i)}(u_m;z_n)$. As it holds that, by definition, $\nabla G_j^{(i)}(u_m;z_n) = \nabla H_{2j-1}^{(i+1)}(u_m) - \nabla H_{2j}^{(i+1)}(z_n-u_m)$, we need to further estimate $\nabla H_{2j-1}^{(i+1)}(u_m)$ and $\nabla H_{2j}^{(i+1)}(z_n-u_m)$.

This is done using the following properties relating the different gradients of $H_j^{(i)}$. They are direct consequences of the envelop theorem (see Lemma 8) because

$$H_j^{(i)}(u) = \max_{\|x\|_1=u} F_j^{(i)}(x) = \max_{v\in[0,u]} H_{2j-1}^{(i+1)}(v) + H_{2j}^{(i+1)}(u-v).$$

As a consequence, if $v^\star$ denotes the point where the maximum is reached and it belongs to $(0,u)$, then

$$\nabla H_j^{(i)}(u) = \nabla H_{2j-1}^{(i+1)}(v^\star) = \nabla H_{2j}^{(i+1)}(u-v^\star). \tag{2}$$

If $v^\star = 0$, then $\nabla H_j^{(i)}(u) = \nabla H_{2j}^{(i+1)}(u-v^\star)$, *i.e.,* the first inequality might only be an inequality. On the other hand, if $v^\star = u$, then $\nabla H_j^{(i)}(u) = \nabla H_{2j-1}^{(i+1)}(v^\star)$. This is the central tool that will let us compute the gradients of all the nodes of the tree, from the leaves to the root.

Thanks to the envelop theorem and Equation (2) we are able to compute the gradients $\nabla G_j^{(i)}(v;u)$ for all nodes in the tree. This is done recursively by imbricating dichotomies.

The goal of the binary searches $\mathcal{D}_{2j-1}^{(i+1)}(v)$ and $\mathcal{D}_{2j}^{(i+1)}(u-v)$ is to compute an approximate value of $\nabla G_j^{(i)}(v;u)$. Indeed we have

$$\nabla G_j^{(i)}(v;u) = \nabla H_{2j-1}^{(i+1)}(v) - \nabla H_{2j}^{(i+1)}(u-v),$$

and to compute $H_{2j-1}^{(i+1)}(v)$ (respectively $\nabla H_{2j}^{(i+1)}(u-v)$) we need to run the binary search $\mathcal{D}_{2j-1}^{(i+1)}(v)$ (respectively $\mathcal{D}_{2j}^{(i+1)}(u-v)$). Let us denote by $\widehat{\nabla} G_j^{(i)}(v;u)$ the approximate value of $\nabla G_j^{(i)}(v;u)$

computed at the end of the binary searches $\mathcal{D}_{2j-1}^{(i+1)}(v)$ and $\mathcal{D}_{2j}^{(i+1)}(u-v)$, that compute themselves $\widehat{\nabla} H_{2j-1}^{(i+1)}(v)$, approximation of $\nabla H_{2j-1}^{(i+1)}(v)$ and $\widehat{\nabla} H_{2j}^{(i+1)}(u-v)$, approximation of $\nabla H_{2j}^{(i+1)}(u-v)$.

The envelop theorem gives that $\nabla H_{2j-1}^{(i+1)}(v) = \nabla H_{4j-3}^{(i+2)}(w^\star) = \nabla H_{4j-2}^{(i+2)}(v-w^\star)$ where $w^\star = \text{argmax} G_{2j-1}^{(i+1)}(w;v)$. Therefore in order to compute $\widehat{\nabla} H_{2j-1}^{(i+1)}(v)$ we run the binary search $\mathcal{D}_{2j-1}^{(i+1)}(v)$ that aims at maximizing the function $\left(w \mapsto G_{2j-1}^{(i+1)}(w;v)\right)$. At iteration N of $\mathcal{D}_{2j-1}^{(i+1)}(v)$, we have

$$|\nabla G_{2j-1}^{(i+1)}(w_N;v)| = |\nabla H_{4j-3}^{(i+2)}(w_N) - \nabla H_{4j-2}^{(i+2)}(v-w_N)|.$$

We use the following estimate for $\nabla H_{2j-1}^{(i+1)}(v)$:

$$\widehat{\nabla} H_{2j-1}^{(i+1)}(v) \doteq \frac{1}{2}\left(\nabla H_{4j-3}^{(i+2)}(w_N) + \nabla H_{4j-2}^{(i+2)}(v-w_N)\right).$$

Since $w^\star \in (w_N, v-w_N)$ (or $(v-w_N, w_N)$), we have that

$$|\widehat{\nabla} H_{2j-1}^{(i+1)}(v) - \nabla H_{2j-1}^{(i+1)}(v)| \leq \frac{1}{2}|\nabla G_{2j-1}^{(i+1)}(w_N;v)|.$$

Consequently we can say that with high probability,

$$\nabla G_j^{(i)}(v;u) \in \left[\widehat{\nabla} G_j^{(i)}(v;u) - \alpha, \widehat{\nabla} G_j^{(i)}(v;u) + \alpha\right]$$

where

$$\alpha = \frac{1}{2}\left(|\nabla G_{2j-1}^{(i+1)}(w_N;v)| + |\nabla G_{2j}^{(i+1)}(v-w_N;v)|\right).$$

In order to be sure that the algorithm does not make an error on the sign of $\nabla G_j^{(i)}(v;u)$ (as in Section 3) we have to run the binary searches $\mathcal{D}_{2j-1}^{(i+1)}(v)$ and $\mathcal{D}_{2j}^{(i+1)}(u-v)$ until $0 \notin \left[\widehat{\nabla} G_j^{(i)}(v;u) - \alpha, \widehat{\nabla} G_j^{(i)}(v;u) + \alpha\right]$ which is the case as soon as $\alpha < |\nabla G_j^{(i)}(v;u)|$. Therefore we decide to stop the binary $\mathcal{D}_{2j-1}^{(i+1)}(v)$ when $|\nabla G_{2j-1}^{(i+1)}(w_N;v)| < |\nabla G_j^{(i)}(v;u)|$ and to stop the binary $\mathcal{D}_{2j}^{(i+1)}(u-v)$ when $|\nabla G_{2j}^{(i+1)}(v-w_N;v)| < |\nabla G_j^{(i)}(v;u)|$.

This leads to the following lemma:

Lemma 24 *During the binary search $\mathcal{D}_{2j-1}^{(i+1)}(v)$ we have, for all point w tested by this binary search,*

$$|\nabla G_{2j-1}^{(i+1)}(w;v)| \geq |\nabla G_j^{(i)}(v)|.$$

And during the binary search $\mathcal{D}_{2j}^{(i+1)}(v)$ we have, for all point w tested by this binary search,

$$|\nabla G_{2j}^{(i+1)}(v-w;v)| \geq |\nabla G_j^{(i)}(v)|.$$

D.2. Analysis of the Algorithm

The rates of convergence of Theorem 7 are obtained by introducing a decomposition of the overall regret incurred by each binary search recursively run by the algorithm.

Definition 25 *We define $R_j^{(i)}(v)$ the regret induced by the binary search $\mathcal{D}_j^{(i)}(v)$ as the regret suffered when optimizing the function $\left(w \mapsto G_j^{(i)}(w;v)\right)$.*

This notion of subregret is crucial for our induction since the regret of the algorithm after T samples satisfies $R(T) = R_0^{(0)}(1)/T$.

The main ingredient of the proof of Theorem 7 is the following Proposition which gives a bound on the subregret $R_j^{(i)}(v)$ depending on the subregrets below it.

Proposition 26 *If $\mathcal{D}_j^{(i)}(v)$ is a node at distance p from the bottom of the binary tree we have:*

$$R_j^{(i)}(v) \leq \sum_{r=1}^{r_{\max}} 8\log(2T/\delta)\frac{\left|g_j^{(i)}(w_r;v)\right|}{\left|\nabla g_j^{(i)}(w_r;v)\right|^2}\log(T)^p + R_{2j-1}^{(i+1)}(w_r) + R_{2j}^{(i+1)}(v-w_r).$$

Where $r_{\max}$ the number of different samples of $\mathcal{D}_j^{(i)}(v)$ and $g_j^{(i)}(\cdot\,;v) \doteq G_j^{(i)}(\cdot\,;v) - G_j^{(i)}(z^{\star}{}_j^{(i)}(v);v)$. We have also $r_{\max} \leq \log_2(T)$.

This proposition is a direct consequence of the following two lemmas that are proven in Appendix E: Lemma 27 gives an expression to compute the subregret $R_j^{(i)}(v)$ and Lemma 28 gives a bound on the number of samples needed to compute $\nabla G_j^{(i)}(w;v)$ at a given precision.

Lemma 27 *The subregret $R_j^{(i)}(v)$ verifies*

$$R_j^{(i)}(v) = \sum_{t=1}^{T_j^{(i)}(v)} \left(\left|G_j^{(i)}(z_j^{(i)}(t);v) - G_j^{(i)}(z^{\star}{}_j^{(i)}(v);v))\right|\right) + \sum_{z\in\{z_j^{(i)}(t),t=1,\dots,T_j^{(i)}(v)\}} R_{2j-1}^{(i+1)}(z) + R_{2j}^{(i+1)}(v-z)$$

where $z^{\star}{}_j^{(i)}(v)$ is the point where $G_j^{(i)}(\cdot\,;v)$ reaches its maximum and where the successive points tested by the binary search $\mathcal{D}_j^{(i)}(v)$ are the (not necessarily distinct) $z_j^{(i)}(t)$.

Lemma 28 *A point w tested by the binary search $\mathcal{D}_j^{(i)}(v)$ has to be sampled at most a number of times equal to*

$$8\log(2T/\delta)\frac{\log(T)^p}{\left|\nabla G_j^{(i)}(w;v)\right|^2},$$

where p is the distance of the node $\mathcal{D}_j^{(i)}(v)$ to the bottom of the binary tree: $p = \log_2(K) - 1 - i$.

Finally it now remains to control the different ratios $\left|g_j^{(i)}(w_r;v)\right| / \left|\nabla g_j^{(i)}(w_r;v)\right|^2$, using the Łojasiewicz inequality and techniques similar to the case of $K = 2$. The main difference is the binary tree we construct that imbricates binary searches. The overall idea is that each layer of that tree adds a multiplicative factor of $\log(T)$.

D.3. Proof of Theorem 7 with $\beta > 2$

Proof Let us first bound a sub-regret $R_j^{(i)}(v)$ for $i \neq 0$. Proposition 26 gives with p the distance from $\mathcal{D}_j^{(i)}(v)$ to the bottom of the tree,

$$R_j^{(i)}(v) \leq \sum_{m=1}^{\log_2(T)} 8\log(2T/\delta)\frac{\left|g_j^{(i)}(w_m;v)\right|}{\left|\nabla g_j^{(i)}(w_m;v)\right|^2}\log_2(T)^p + R_{2j-1}^{(i+1)}(w_m) + R_{2j}^{(i+1)}(v-w_m).$$

For the sake of simplicity we will note $g = g_j^{(i)}(\cdot\,;v)$, and we will begin by bounding

$$R = \log(T)^p \sum_{m=1}^{\log_2(T)} \frac{|g(w_m)|}{|\nabla g(w_m)|^2}.$$

We use the Łojasiewicz inequality to obtain that $|g(w_m)| \leq c|\nabla g(w_m)|^\beta$. This gives

$$R \leq c^{2/\beta}\log_2(T)^p \sum_{m=1}^{\log_2(T)} |g(w_m)|^{1-2/\beta}$$

We are now in a similar situation as in the proof of Theorem 5 in the case where $\beta > 2$. Using the fact that $|g(w_m)| \leq L2^{-m}$, we have

$$R \leq \frac{1}{1-2^{2/\beta-1}} c^{2/\beta} L^{1-2/\beta}\log_2(T)^p.$$

Let us note $C \doteq \frac{1}{1-2^{2/\beta-1}} c^{2/\beta} L^{1-2/\beta}$. We have $R \leq C\log_2(T)^p$.

We use now Proposition 26 which shows that

$$R_j^{(i)}(v) \leq 8\log(2T/\delta)\cdot C\log_2(T)^p + \sum_{m=1}^{\log_2(T)} R_{2j-1}^{(i+1)}(w_m) + \sum_{m=1}^{\log_2(T)} R_{2j}^{(i+1)}(v-w_m).$$

Let us now define the sequence $A_p = 2A_{p-1}+1$ for $p \geq 1$, and $A_0 = 1$. The bound we have just shown let us show by recurrence that

$$R_j^{(i)}(v) \leq 8\log(2T/\delta)\cdot A_p C\log(T)^p.$$

Lemma 30 shows that $A_p = 2^{p+1}-1 \leq 2^{p+1}$. Moreover for $i=0$, we have $p = \log_2(K)-1$. Consequently for $i=0$, $A_p \leq K$.

With the choice of $\delta = 2/T^2$ we have finally that

$$R(T) = \frac{R_1^{(0)}(1)}{T} \leq 8\cdot KC\frac{\log(T)^{\log_2(K)}}{T} \lesssim \frac{1}{1-2^{2/\beta-1}} c^{2/\beta} L^{1-2/\beta} K\frac{\log(T)^{\log_2(K)}}{T}.$$

■

D.4. Proof of Theorem 7 with $\beta = 2$

Proof

Let us first bound a sub-regret $R_j^{(i)}(v)$ for $i \neq 0$. Proposition 26 gives with p the distance from $\mathcal{D}_j^{(i)}(v)$ to the bottom of the tree,

$$R_j^{(i)}(v) \leq \sum_{m=1}^{\log_2(T)} 8\log(2T/\delta)\frac{\left|g_j^{(i)}(w_m;v)\right|}{\left|\nabla g_j^{(i)}(w_m;v)\right|^2}\log_2(T)^p + R_{2j-1}^{(i+1)}(w_m) + R_{2j}^{(i+1)}(v-w_m).$$

For the sake of simplicity we will note $g = g_j^{(i)}(\cdot\,;v)$ and and we will begin by bounding

$$R = \sum_{m=1}^{\log(T)} \frac{|g(w_m)|}{|\nabla g(w_m)|^2}\log(T)^p.$$

Łojasiewicz inequality gives $|g(w_m)| \leq c|\nabla g(w_m)|^2$, leading to

$$R \leq \sum_{m=1}^{\log(T)} c\log(T)^p \leq c\log(T)^{p+1}.$$

An immediate recurrence gives that, as in the case where $\beta > 2$,

$$R_j^{(i)}(v) \leq 8A_p c\log(2T/\delta)\log(T)^{p+1}.$$

And finally we have, noting $g \doteq g_1^{(0)}(\cdot\,;1)$ and $p = \log_2(K) - 1$

$$R_1^{(0)}(1) \leq 8A_p c\log(2T/\delta)\log(T)^{\log_d(K)}.$$

Giving finally, with the choice $\delta = 2/T^2$ and since $A_p \leq K$ for $p = \log_2(K) - 1$,

$$R(T) = 8A_p c\log(2T/\delta)\frac{R}{T} \leq 24cK\frac{\log(T)^{\log_2(K)+1}}{T}.$$

■

D.5. Proof of Theorem 7 with $\beta < 2$

Proof Let us first bound a sub-regret $R_j^{(i)}(v)$ for $i \neq 0$. Proposition 26 gives with p the distance from $\mathcal{D}_j^{(i)}(v)$ to the bottom of the tree,

$$R_j^{(i)}(v) \leq \sum_{m=1}^{\log(T)} 8\log(2T/\delta)\frac{\left|g_j^{(i)}(w_m;v)\right|}{\left|\nabla g_j^{(i)}(w_m;v)\right|^2}\log(T)^p + R_{2j-1}^{(i+1)}(w_m) + R_{2j}^{(i+1)}(v-w_m).$$

For the sake of simplicity we will note $g = g_j^{(i)}(\cdot\,; v)$ and we will begin by bounding

$$R = \sum_{m=1}^{\log_2(T)} \frac{|g(w_m)|}{|\nabla g(w_m)|^2} \log(T)^p.$$

Łojasiewicz inequality gives $|g(w_m)| \leq c|\nabla g(w_m)|^\beta$, leading to

$$R \leq \sum_{m=1}^{\log_2(T)} c|\nabla g(w_m)|^{\beta-2} \log_2(T)^p.$$

We want to prove by recurrence that, with $p = \log_2(K) - 1 - i$ and A_p defined in Section D.3.

$$R_j^{(i)}(v) \leq 8\log(2T/\delta) c A_p \sum_{r=1}^{r_{\max}} \left|\nabla G_j^{(i)}(w_r; v)\right|^{\beta-2} \log_2(T)^p. \tag{3}$$

The result is true for $p = 0$ using what has be done previously. Suppose that it holds at level $i+1$ in the tree. Then, Proposition 26 shows that

$$\begin{aligned} R_j^{(i)}(v) &\leq \sum_{r=1}^{r_{\max}} 8\log(2T/\delta) \frac{\left|g_j^{(i)}(w_r; v)\right|}{\left|\nabla G_j^{(i)}(w_r; v)\right|^2} \log(T)^p + R_{2j-1}^{(i+1)}(w_r) + R_{2j}^{(i+1)}(v - w_r) \\ &\leq 8\log(2T/\delta)\Bigg(\log_2(T)^p \sum_{r=1}^{r_{\max}} c\left|\nabla G_j^{(i)}(w_r)\right|^{\beta-2} + \sum_{r=1}^{r_{\max}} cA_{p-1} \sum_{s=1}^{s_{\max}} \left|\nabla G_{2j-1}^{(i+1)}(x_s; w_r)\right|^{\beta-2} \log_2(T)^{p-1} \\ &\qquad + \sum_{r=1}^{r_{\max}} cA_{p-1} \sum_{s=1}^{s_{\max}} \left|\nabla G_{2j}^{(i+1)}(\tilde{x}_s; v - w_r)\right|^{\beta-2} \log_2(T)^{p-1}\Bigg). \end{aligned}$$

We have noted by x_s and $\tilde{x}_s$ the points tested by the binary searches $\mathcal{D}_{2j-1}^{(i+1)}(w_r)$ and $\mathcal{D}_{2j}^{(i+1)}(v - w_r)$ and $s_{\max} \leq \log_2(T)$ the number of points tested by those binary searches. We now use the fact that $\beta - 2 < 0$ and Lemma 24 shows that $\left|\nabla G_{2j-1}^{(i+1)}(x_s; w_m)\right| \geq \left|\nabla G_j^{(i)}(w_r)\right|$, giving

$$R_j^{(i)}(v) \leq (1 + 2A_{p-1}) c \cdot 8\log(2T/\delta) \sum_{r=1}^{r_{\max}} \left|\nabla G_j^{(i)}(w_r; v)\right|^{\beta-2} \log_2(T)^p,$$

proving Equation (3). And finally we have, as in the proof of Theorem 5, noting $g \doteq g_1^{(0)}(\cdot\,; 1)$,

$$R_1^{(0)}(1) \leq Kc \cdot 8\log(2T/\delta) \sum_{r=1}^{r_{\max}} |\nabla g(u_r)|^{\beta-2} \log(T)^{\log_2(K)-1}.$$

We note now $g_r \doteq |\nabla g(u_r)|$ and we have the constraint, with $T' = \dfrac{T}{8\log(2T/\delta)\log(T)^{\log_2(K)-1}}$

$$T' = \sum_{r=1}^{r_{\max}} \frac{1}{g_r^2}.$$

We want to maximize $R \doteq \sum_{r=1}^{r_{\max}} g_r^{\beta-2}$ under the above constraint.

In order to do that we introduce the following Lagrangian function:

$$\mathcal{L} : (g_1, \dots, g_{r_{\max}}, \lambda) \mapsto \sum_{r=1}^{r_{\max}} g_r^{\beta-2} + \lambda \left(T' - \sum_{r=1}^{r_{\max}} \frac{1}{g_r^2} \right).$$

The Karush-Kuhn-Tucker theorem gives

$$\begin{aligned}
0 &= \frac{\partial \mathcal{L}}{\partial g_r}(g_1, \dots, g_{r_{\max}}, \lambda) \\
0 &= (\beta - 2) g_r^{\beta-3} + \lambda \left(2 g_r^{-3} \right) \\
0 &= (\beta - 2) g_r^{\beta} + 2\lambda \\
g_r &= \left(\frac{2\lambda}{2-\beta} \right)^{1/\beta}.
\end{aligned}$$

The expression of T' gives

$$\begin{aligned}
T' &= \sum_{r=1}^{r_{\max}} g_r^{-2} \\
T' &= \sum_{r=1}^{r_{\max}} \left(\frac{2\lambda}{2-\beta} \right)^{-2/\beta} \\
\lambda^{-2/\beta} &= \frac{T'}{\sum_{r=1}^{r_{\max}} (1-\beta/2)^{2/\beta}} \\
\lambda &= T'^{-\beta/2} r_{\max}^{\beta/2} (1-\beta/2).
\end{aligned}$$

We can now bound R:

$$\begin{aligned}
R &\leq \sum_{r=1}^{r_{\max}} g_r^{\beta-2} \\
&\leq \sum_{r=1}^{r_{\max}} \left(\frac{2\lambda}{2-\beta} \right)^{1-2/\beta} \\
&\leq r_{\max} (1-\beta/2)^{2/\beta-1} \lambda^{1-2/\beta} \\
&\leq r_{\max} (1-\beta/2)^{2/\beta-1} \left(T'^{-\beta/2} r_{\max}^{\beta/2} (1-\beta/2) \right)^{1-2/\beta} \\
&\leq r_{\max}^{\beta/2} T'^{1-\beta/2}.
\end{aligned}$$

Now we use the fact that $R(T) = \dfrac{R_1^{(0)}(1)}{T}$ and $R_1^{(0)}(1) \leq Kc \cdot 8 \log(2T/\delta) \log(T)^{\log_2(K)-1} R$.

Taking $\delta = 2/T^2$, we have $\log(2T/\delta) = 3\log(T)$. We have, since $r_{\max} \leq \log(T)$,

$$\begin{aligned}
R(T) &\leq \frac{1}{T} Kc \cdot 8 \log(2T/\delta) \log(T)^{\log_2(K)-1} R \\
&\leq \frac{24Kc}{T} \log(T)^{\log_2(K)} R \\
&\leq \frac{24Kc}{T} \log(T)^{\log_2(K)} r_{\max}^{\beta/2} T'^{1-\beta/2} \\
&\leq \frac{24Kc}{T} \log(T)^{\log_2(K)} \log(T)^{\beta/2} \left(\frac{T}{24 \log(T)^{\log_2(K)}} \right)^{1-\beta/2} \\
&\leq 24^{\beta/2} Kc \left(\frac{\log(T)^{\log_2(K)+1}}{T} \right)^{\beta/2}.
\end{aligned}$$

■

Appendix E. Proof of Technical and Simple Results

E.1. A simple lemma

We start with a simple lemma that will be useful in the following proofs.

Lemma 29 *For all $x \in [0,1]$, we have $|g(x)| \leq |g'(x)||x^\star - x|$.*

Proof Since g is non-positive and $g(x^\star) = 0$, we have for all $x \in [0,1]$:

$$|g(x)| = -g(x) = g(x^\star) - g(x) = \int_x^{x^\star} g'(y)\, \mathrm{d}y.$$

Let us distinguish two cases depending on $x < x^\star$ or $x > x^\star$.

- $x < x^\star$: since g' is non-increasing (because g is concave) we have for all $y \in [x, x^\star]$, $g'(y) \leq g'(x)$ and therefore

  $$|g(x)| \leq |x^\star - x| g'(x) = |g'(x)||x^\star - x|.$$

  We have indeed $g'(x) \geq 0$ because $x < x^\star$.

- $x > x^\star$: similarly we have for all $y \in [x^\star, x]$, $g'(y) \geq g'(x)$ and therefore

  $$|g(x)| = -\int_{x^\star}^{x} g'(y)\, \mathrm{d}y \leq |x^\star - x|(-g'(x)) = |g'(x)||x^\star - x|.$$

  We have indeed $g'(x) \leq 0$ because $x > x^\star$ and g' non-increasing.

■

E.2. Proof of Lemma 3

Proof This lemma is just a consequence of Hoeffding inequality. Indeed, it implies that, at stage $n \in \mathbb{N}$,

$$\mathbb{P}\Big\{\Big|\frac{1}{n}\sum_{t=1}^{n} X_t - x\Big| \geq \sqrt{\frac{2\log(\frac{2T}{\delta})}{n}}\Big\} \leq \frac{\delta}{T}\,,$$

thus with probability at least $1-\delta$, x belongs to $\Big[\frac{1}{N_x}\sum_{t=1}^{N_x} X_t \pm \sqrt{\frac{2\log(\frac{2T}{\delta})}{N_x}}\Big]$ and the sign of x is never mistakenly determined.

On the other hand, at stage N_x, it holds on the same event that $\frac{1}{N_x}\sum_{t=1}^{N_x} X_t$ is $\frac{x}{2}$-close to x, thus 0 no longer belongs to the interval $\Big[\frac{1}{N_x}\sum_{t=1}^{N_x} X_t \pm \sqrt{\frac{2\log(\frac{2T}{\delta})}{N_x}}\Big]$. ■

E.3. A simple arithmetic lemma

We state and prove here a simple arithmetic lemma useful in the proof of Theorem 7.

Lemma 30 *Let $(u_n)_{n\in\mathbb{N}} \in \mathbb{N}^{\mathbb{N}}$ defined as follows: $u_0 = 1$ and $u_{n+1} = 2u_n + 1$. Then*

$$\forall n \in \mathbb{N},\ u_n = 2^{n+1} - 1.$$

Proof Let consider the sequence $v_n = u_n + 1$. We have $v_0 = 2$ and $v_{n+1} = 2v_n$. Consequently $v_n = 2 \cdot 2^n = 2^{n+1}$. ■

E.4. Proof of Proposition 26

Proof The statement of Proposition 26 is a restatement of Lemma 27 using the fact that each different point of the binary search $\mathcal{D}_j^{(i)}(v)$ is sampled a number of times equal to $8\log(2T/\delta)\dfrac{\log(T)^p}{\Big|\nabla G_j^{(i)}(w;v)\Big|^2}$ thanks to Lemma 28. The fact that $r_{\max} \leq \log_2(T)$ comes from the fact that running a binary search to a precision smaller than $1/LT$ does not give improved bound on the regret since the reward functions are L-Lipschitz continuous. Therefore the binary searches are stopped after more than $\log_2(T)$ samples. ■

E.4.1. Proof of Lemma 27

Proof The regret of the binary search $\mathcal{D}_j^{(i)}(v)$ is the sum for all steps $t \in [T_j^{(i)}(v)]$ of the sum of two terms: the difference of the function values of $G_j^{(i)}(\cdot\,;v)$ between the optimal value $z^{\star(i)}_j(v)$ and $z_j^{(i)}(t)$ and the sub-regrets $R_{2j-1}^{(i+1)}(z_j^{(i)}(t))$ and $R_{2j}^{(i+1)}(v - z_j^{(i)}(t))$ of the binary searches that are the children of $\mathcal{D}_j^{(i)}(v)$. ■

E.4.2. PROOF OF LEMMA 28

Proof The binary search $\mathcal{D}_j^{(i)}(v)$ aims at minimizing the function $(w \mapsto G_j^{(i)}(w;v))$. Let us note $w_1, \dots, w_m, \dots$ the values that are tested by this binary search. During the binary search the signs of the values of $\nabla G_j^{(i)}(w_m;v)$ are needed. In order to compute them the algorithm runs sub-binary searches (unless $\mathcal{D}_j^{(i)}(v)$ is a leaf) $\mathcal{D}_{2j-1}^{(i+1)}(w_m)$ and $\mathcal{D}_{2j}^{(i+1)}(v-w_m)$.

Let us now prove the result by recurrence on the distance p of $\mathcal{D}_j^{(i)}$ to the closest leaf of the tree.

- $p=0$: $\mathcal{D}_j^{(i)}$ is a leaf. The point w_m needs to be sampled $8\log(2T/\delta)/\left|\nabla g_j^{(i)}(w_m)\right|^2$ (this has been shown in Section 3).
- $p \in \mathbb{N}^*$: the point w_m has to be sampled a number of times equal to the number of iterations of $\mathcal{D}_{2j-1}^{(i+1)}(w_m)$ and $\mathcal{D}_{2j}^{(i+1)}(v-w_m)$. Let us therefore compute the number of samples used by $\mathcal{D}_{2j-1}^{(i+1)}(w_m)$. This binary search is at distance $p-1$ of the closest leaf. Therefore by hypothesis recurrence each point x_k will be sampled a number of times equal to

$$N_k = 8\log(2T/\delta)\frac{\log(T)^{p-1}}{\left|\nabla G_{2j-1}^{(i+1)}(x_k)\right|^2}.$$

Now Lemma 24 shows that $\left|\nabla G_{2j-1}^{(i+1)}(x_k)\right| \geq \left|\nabla G_j^{(i)}(w_m)\right|$. This gives

$$N_k \leq 8\log(2T/\delta)\frac{\log(T)^{p-1}}{\left|\nabla G_j^{(i)}(w_m)\right|}.$$

The same reasoning applies for the binary search $\mathcal{D}_{2j}^{(i+1)}(v-w_m)$, which is run in parallel to $\mathcal{D}_{2j-1}^{(i+1)}(w_m)$. Since there are at most $\log_2(T)$ different points x_k that are tested during the binary search $\mathcal{D}_{2j-1}^{(i+1)}(w_m)$, we have a final number of iterations for w_m which is

$$8\log(2T/\delta)\frac{\log(T)^{p}}{\left|\nabla G_j^{(i)}(w_m)\right|}.$$

This proves the result for the step p.

- Finally the recurrence is complete and the result is shown.

■

Appendix F. Experiments

In this section, we illustrate the performances of our algorithm on generated data with $K=2$ resources. We have considered different possible values for the parameter $\beta \in [1,\infty)$.

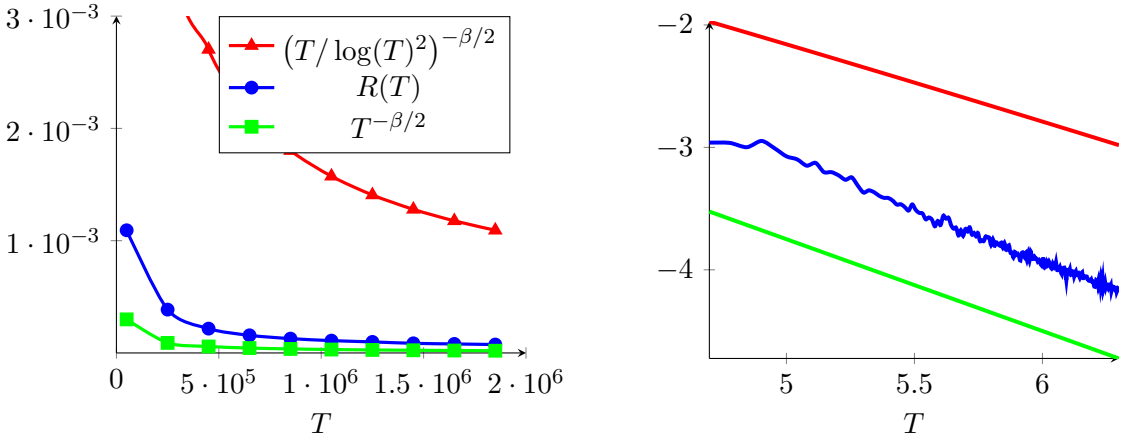
In the case where $\beta=2$ we have considered the following functions:

$$f_1 : x \mapsto \frac{5}{6} - \frac{5}{48}(2-x)^3 \quad \text{and} \quad f_2 : x \mapsto \frac{6655}{384} - \frac{5}{48}\left(\frac{11}{5}-x\right)^3$$

such that $g(x) = -(x - 0.4)^2$. g verifies the Łojasiewicz inequality with $\beta = 2$ and the functions f_1 and f_2 are concave, non-decreasing and take value 0 at 0.

We have computed the cumulated regret of our algorithm in various settings corresponding to different values of β and we have plotted the two references rates: the lower bound $T^{-\beta/2}$ (even if the functions considered in our examples are not those used to prove the lower bound), and the upper bound $(T/\log^2(T))^{-\beta/2}$.

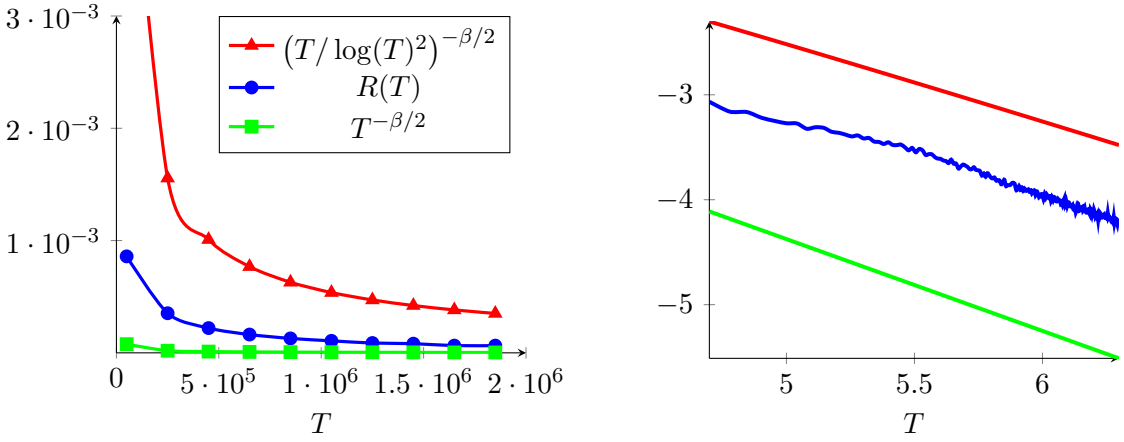
Our experimental results on Figures 1, 2, 3 and 4 indicate that our algorithm has the correct expected behavior, as its regret is "squeezed" between $T^{-\beta/2}$ and $(T/\log^2(T))^{-\beta/2}$ for $\beta \leq 2$ and between T^{-1} and $\log(T)/T$ for $\beta \geq 2$. Moreover, the $\log - \log$ scale also illustrates that $-\beta/2$ is indeed the correct speed of convergence for functions that satisfy the Łojasiewicz inequality with respect to $\beta \in [1, 2]$.



(a) Regret as a function of T

(b) Regret in $\log - \log$ scale

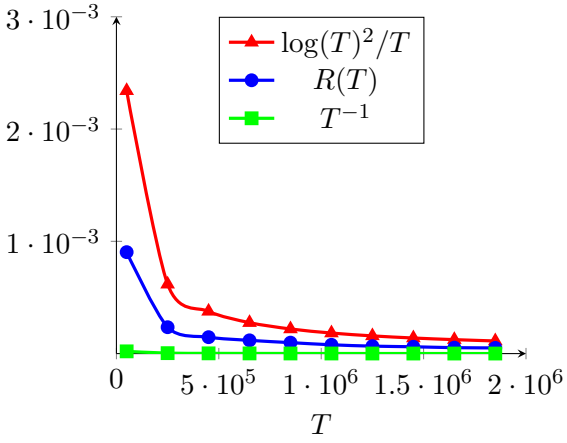
Figure 1: Regret, Upper-bound and Lower bound for $\beta = 1.5$



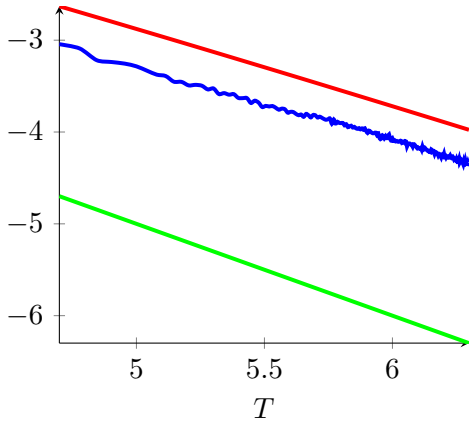
(a) Regret as a function of T

(b) Regret in $\log - \log$ scale

Figure 2: Regret, Upper-bound and Lower bound for $\beta = 1.75$

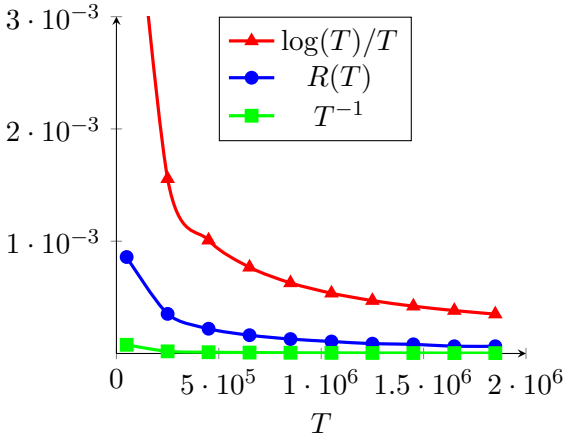


(a) Regret as a function of T

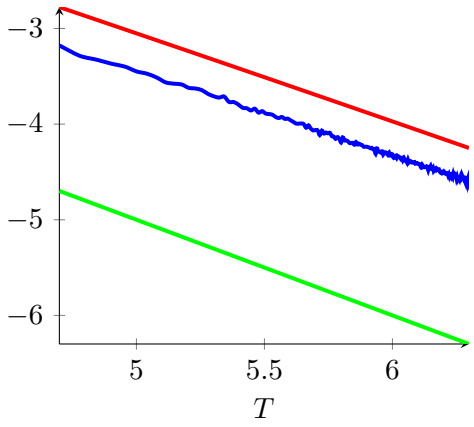


(b) Regret in $\log - \log$ scale

Figure 3: Regret, Upper-bound and Lower bound for $\beta = 2$



(a) Regret as a function of T



(b) Regret in $\log - \log$ scale

Figure 4: Regret, Upper-bound and Lower bound for $\beta = 2.5$

We plot in Figure 5 the regret curves obtained for different values of the parameter β. This validates the fact that the convergence rates increase with the value of β as proved theoretically.

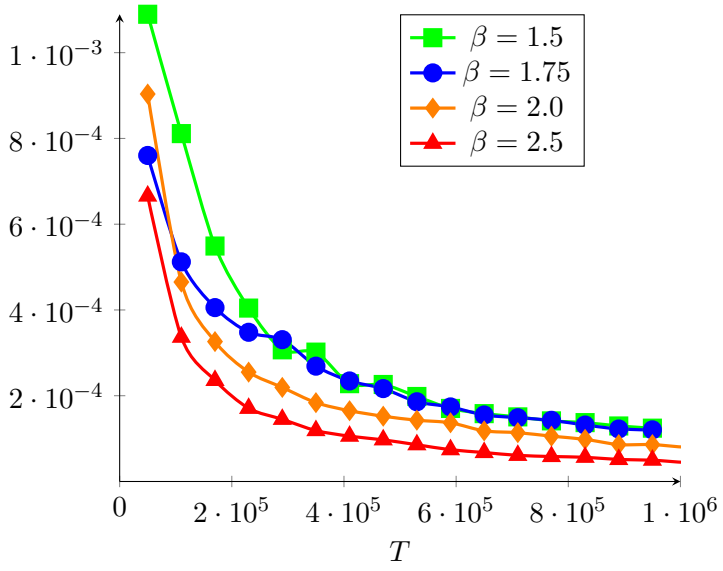


Figure 5: Regret as a Function of T for different values of β