Adaptive multi-fidelity optimization with fast learning rates

Côme Fiegel
École Normale Supérieure, Paris
Inria Lille

Victor Gabillon
Huawei R&D, UK

Michal Valko
Inria Lille

Abstract

In multi-fidelity optimization, we have access to biased approximations of varying costs of the target function. In this work, we study the setting of optimizing a locally smooth function with a limited budget Λ, where the learner has to make a trade-off between the cost and the bias of these approximations. First, we prove lower bounds for the simple regret under different assumptions on the fidelities, based on a cost-to-bias function. We then present the Kometo algorithm which achieves, with additional logarithmic factors, the same rates *without* any knowledge of the function smoothness and fidelity assumptions, and improves prior results. Finally, we empirically show that our algorithm outperforms prior multi-fidelity optimization methods without the knowledge of problem-dependent parameters.

1 Introduction

In multi-fidelity optimization (Cutler et al., 2014; Huang et al., 2006; Kandasamy et al., 2016c, 2017), the learner actively optimizes a function but only observes, at each of the rounds, biased values of that function. The learner can *pay* to reduce the bias of the observed function values. The smaller the bias the higher the cost, urging the learner to carefully allocate its total cost budget Λ on the fly.

We consider the case of *derivative-free optimization* where no gradient information is available (Matyas, 1965). This is of great interest for the multiple applications in which it is either difficult to access, compute, or even define gradients (Nesterov and Spokoiny, 2017). Using only zero-order information, derivative-free optimization addresses optimising over functions that are not differentiable, non-continuous, or non-smooth. Moreover, there are known methods that work without knowing the smoothness parameters (ν, ρ) of the function (Auer et al., 2007; Kleinberg et al., 2008; Grill et al., 2015).

Derivative-free multi-fidelity optimization is useful in particular for the hyper-parameters tuning of *complex* machine learning models, where each evaluation of the model is costly such as tokamak simulators. However, the mapping between the hyper-parameter and performance of the learned model can be highly non-convex and non-smooth. Moreover, training a model, given the hyper-parameters can be expensive and time-consuming (Sen et al., 2018). In a situation, where computation or time are constrained by a budget, these constraints prevent us from carefully evaluating the qualities of all the models generated from a continuous set of hyper-parameters. Then, given one fixed set of hyper-parameters, the bias of the estimation of the quality of fully-trained model is a (decreasing) function ζ of the amount of computation resource spent training the model. Ultimately, we would expect this bias to be zero if the model is trained until convergence. However, the bias function ζ is a function that depends on the type of trained models and that is a priori unknown in applications .

The most related approach for the considered setting is the MFPDOO algorithm of Sen et al. (2018). In order to provide theoretical guarantees for MFPDOO, ζ is either assumed to be known or some parametric assumptions on ζ are made and the parameters are estimated online. However, knowing ζ or its parametric family is unrealistic.

In this paper, we propose a new method called Kometo that adapts to the unknown ζ and the unknown smoothness parameters (ν, ρ). Our analysis is more general than the analysis of Sen et al. (2018) and provides a broader and finer set of behaviors of the cost-to-bias function. This allows us to provide a characterisation of the complexity of the problem by providing the first regret lower bounds in multi-fidelity optimization. We

Proceedings of the 23$^{\text{rd}}$International Conference on Artificial Intelligence and Statistics (AISTATS) 2020, Palermo, Italy. PMLR: Volume 108. Copyright 2020 by the author(s).

also show that KOMETO obtains rates that match the ones of our lower bounds and improves upon the rates of MFPDOO *while dropping the assumptions of knowing the bias function ζ in advance.*

Related work Among the large work on derivative-free optimization, we focus on algorithms that perform well under *minimal* assumptions as well as minimal knowledge of the function. Under *weak/local* smoothness around one global maximum (Auer et al., 2007; Kleinberg et al., 2008; Bubeck et al., 2011), some algorithms require the knowledge of the local smoothness such as HOO (Bubeck et al., 2011), ZOOMING (Kleinberg et al., 2008), or DOO (Munos, 2011). Among the work relying on an *unknown* local smoothness, SequOOL (Bartlett et al., 2019) improves on SOO (Munos, 2011; Kawaguchi et al., 2016) and represents the state-of-the-art for the deterministic feedback. For the stochastic feedback, StoSOO (Valko et al., 2013) extends SOO for a limited class of functions. POO (Grill et al., 2015) and GPO (Shang et al., 2019) provide more general results. Finally, StroquOOL (Bartlett et al., 2019) matches, up to log factors, the guarantees of SequOOL and GPO for deterministic and stochastic feedback respectively, *without requiring the knowledge of the range of the noise b.*

Multi-fidelity optimization is a well studied setting. Here, we address *online* multi-fidelity optimization. Many of approaches rely on Bayesian models, e.g., Gaussian processes. Zhang et al. (2019) relies on entropic search to find the maximum, while Kandasamy et al. (2016a) adapts GP-UCB (Srinivas et al., 2010) to multi-fidelity setting. Most of these methods need an access to a bias function, while Ghosh et al. (2019) uses the cost of the approximations to estimate its values. Li et al. (2017) obtains good empirical results by trying a lot of configurations at low fidelities and progressively eliminating the less interesting ones while using higher and higher fidelities. Two prior works adapted algorithms working under local smoothness around one global maximum to multi-fidelity settings. First, Sen et al. (2018) adapted POO (Grill et al., 2015) to deterministic multi-fidelity settings and later Sen et al. (2019) made it work under stochastic ones.

Main contributions

- We introduce more general assumptions on the fidelity approximations, based on their cost, while keeping the smoothness assumption on the target function.
- We prove lower bounds of the simple regret under these different assumptions.
- We provide KOMETO, an algorithm that, in deterministic settings, without *any* knowledge on the bias function and the smoothness of the target function, achieves minimax optimal rates for simple regret up to logarithmic factors on all considered assumptions on the fidelities. It improves the previously proven guaranteed rates under local smoothness assumptions of Sen et al. (2018), except in the case $\alpha = 1$ of Assumption 2(a),[1] where it needs additional logarithmic factors. KOMETO comes with important properties:
  - It does *not* assume an access to the target function, only an access to increasingly better approximations, unlike previous algorithms like Sen et al. (2018).
  - It only uses the comparisons of evaluations at the same fidelity level, and not directly the values of the evaluations, which leads to weaker fidelity assumptions and better empirical results.
  - It works in stochastic settings by changing the number of evaluations at higher fidelities.
- We provide synthetic experiments and a hyperparameter tuning experiment to demonstrate the efficiency of KOMETO.

2 Problem settings

In this section, we introduce a generalization of the settings presented by Sen et al. (2018).

We want to optimize a target function $f : \mathcal{X} \to \mathbb{R}$ under a budget $\Lambda \in \mathbb{R}_+$. The evaluation of this target function is done through its fidelity approximations. We thus denote by $Z = [0, 1]$ the fidelity space and by $(f_z)_{z\in Z}$ the fidelity approximations. In particular, $z = 0$ corresponds to the lowest fidelity, while $z = 1$ corresponds to the highest one. We also denote by $\zeta : Z \to \overline{\mathbb{R}_+}$ the unknown bias function, such that there exists a family $(g_z)_{z\in Z}$ of real-valued strictly increasing function with $\|f - g_z \circ f_z\|_\infty \leq \zeta(z)$ for $z \in Z$ (motivations for this hypothesis are explained below). A known cost function $\lambda : Z \to \overline{\mathbb{R}_+}$ indicates the budget used at each evaluation for a given fidelity. We also assume that the algorithm can request, for any $c \geq 1$, a fidelity z_c such that $\lambda(z_c) \leq c$, and we define $\Phi : [1, +\infty[\to \overline{\mathbb{R}_+}$ with $\Phi(c) = \zeta(z_c)$, the cost-to-bias function, which gives for each cost c the minimal bias that one can can be guaranteed for an observation of f. Assumptions 2 are on this function.

At round t, the algorithm makes an evaluation of the function of a point $x_t \in \mathcal{X}$ and at a fidelity $z_t \in Z$ (or at a cost c_t, see above), as long as $\sum_{s=1}^{t} \lambda(z_s) \leq \Lambda$.

[1] hyperbolic decreasing of the cost-to-bias function

The algorithm observes at round t the value $f_{z_t}(x_t)$ in return. The algorithm must finally output a value x_Λ. We then define the simple regret of a policy π for r_Λ as

$$r_\Lambda(\pi) \triangleq \mathbb{E}(\max_{x\in\mathcal{X}} f(x) - f(x_\Lambda))$$

(the expectation is on the randomness of the algorithm). In the rest of this paper, we only aim to minimize this regret, without any constraint on time or space complexity.

Problem setting remarks One of the main aspects our approach is that we *do not assume to have an access to the bias function* ζ. This highlights the fact that our algorithm is fully adaptive, and only needs the cost of each fidelity as an input. Since the bias function is usually unknown in practice, several papers rely on various techniques (e.g., MLE) to guess the values of this function for implementations, but often assume it has a specific form. Given the known results is therefore surprising that our approach gets fast rates, and it does so *without relying on any information on the bias function.*

Moreover we relax the original assumption of Sen et al. (2018) that $\|f - f_z\|_\infty \le \zeta(z)$ for $z \in Z$ and use instead $\|f - g_z \circ f_z\|_\infty \le \zeta(z)$ for $z \in Z$. This lets the f_z approximations be potentially arbitrarily biased with respect to f as long as the ordering in f is approximately kept. Indeed, as g_z are increasing functions, we have that $g_z \circ f_z(x_1) \ge g_z \circ f_z(x_2)$ iif $f_z(x_1) \ge f_z(x_2)$ for any $x_1, x_2 \in \mathcal{X}$. This more general model for example fits in cases where evaluating at lower fidelities (with higher bias) has a great impact on individual feedback, but a low impact on how each different points compare to each other at the same fidelity level. This is for example the case on neural network training, where evaluating with less iterations (lower fidelity) may increase the overall error for every set of hyper-parameters at similar rates. Note that theoretical results will simultaneously hold under both assumption as long as two conditions are met. First, the behavior of our algorithms is not based directly on the (estimated) value of the function f_z but only on comparisons of these estimates of f_z. Second the estimates that are compared are computed from evaluations coming from the same fidelity. This is the case of our algorithm Kometo. Indeed Kometo similarly as SOO (as noted by Munos, 2014) or SequOOL, a *rank-based algorithm.* This means that its behavior is based on the rank of the function evaluations, and not directly on their values. On the contrary the behavior of MFPDOO relies directly on the values in practice when estimating the constant of the parametric model, and would therefore not extend to our general assumption.

Another particularity is that we do not assume that the cost function λ is bounded. We assume quite generally that $\lambda : Z \to \overline{\mathbb{R}_+}$ instead of restricting ourself to having $\lambda : Z \to [0, 1]$ as in Sen et al. (2018). In our scenario, it can happen that some approximations of the function f with low bias are simply too costly for our limited budget. Working under this larger assumption fits better problems in which we can only access feedback from imperfect simulators while the real phenomenon can not be directly evaluated in practice. In such scenario, the MFPDOO Sen et al. (2018) is not usable as it assumes that it directly evaluate the target function f with finite cost during its final cross-validation phase. Our results can also be extended to cases where fidelity space is discrete, by using a piecewise constant cost function.

Finally instead of minimizing the simple regret, the cumulative regret has been also studied in multi-fidelity setting (Kandasamy et al., 2016b), rewarding all accurate evaluations of the target function. However in the present paper we optimize the simple regret as our initial objective is to find the optimum of the target function. The simple regret is adapted to the objectives of Kometo, e.g., hyper-parameter optimization, where we wish to spend the entire budget on *pure exploration.*

3 Assumptions

Our algorithm needs two assumptions, one on the target function (which describes its smoothness) and one on the fidelity approximations (which characterizes how well they approximate the function).

Hierarchical partitioning We use the notion of hierarchical partitioning (Munos, 2011). At every depth $h \ge 0$, $\mathcal{X}$ (potentially multi-dimensional) is partitioned into K^h different cells $(\mathcal{P}_{h,i})_{0\le i\le K^h-1}$. All the cells $(\mathcal{P}_{h,i})_{h,i}$ form a tree, where the root is $P_{0,0} = \mathcal{X}$, and where each cell $\mathcal{P}_{h,i}$ has K children, $(\mathcal{P}_{h+1,Ki+l})_{0\le l\le K-1}$, which form a partition of their parent cell.

We make an assumption on the target function f and the hierarchical partitioning $\mathcal{P}$, identical to the settings of Sen et al. (2018). This following assumption is way weaker than global Lipschitzness and as explained by Grill et al. (2015) is simpler and weaker than assumptions made in previous works (Auer et al., 2007; Munos, 2011).

Assumption 1. *[Assumption on the target function] For one of the global optimum $x^\star$ of f, there exists $\nu > 0$ and $\rho \in]0, 1[$ such that $\forall h \in \mathbb{N}, \forall x \in \mathcal{P}_{h,i^\star_h}, f(x) \ge f(x^\star) - \nu\rho^h$, where $\mathcal{P}_{h,i^\star_h}$ is the cell of depth h containing $x^\star$*

We now define a notion of near-optimality dimension that only depends on the hierarchical partitioning $\mathcal{P}$, and not on a metric.

Definition 1. ***Near-optimality dimension****: For any $\nu > 0$ and $\rho \in \,]0, 1[$, we say that $d \in \mathbb{R}_+$ is a near optimality dimension of f with respect to the partitioning $\mathcal{P}$ and the smoothness parameters (ν, ρ) if*

$$\exists C > 1, \forall h \in \mathbb{N}, N_h(3\nu\rho^h) \leq C\rho^{-dh},$$

where $N_h(\varepsilon)$ is the number of cells $\mathcal{P}_{h,i}$ such that

$$\sup_{x \in \mathcal{P}_{h,i}} f(x) \geq f(x^\star) - \varepsilon.$$

We then define

$$S(\mathcal{P}, \nu, \rho, d, C) \triangleq \{f : X \to \mathbb{R} \,|\, f \text{ has smoothness parameters } (\nu, \rho) \text{ for } \mathcal{P} \text{ and } d \text{ is a near-optimality dimension with associated constant C}\}$$

Note on the near-optimality dimension definition: Grill et al. (2015) define *the* near-optimality dimension as the infimum of the set of d that satisfies this definition (with ν and ρ fixed). However, they then assume that this infimum also satisfies this definition, which is not necessarily true (the set can be of the form $\mathbb{R}_{>0}$ for example). Bartlett et al. (2019) solve this issue by adding an extra dependence on the constant C to get a closed set (fixed parameters are then ν, ρ and C, instead of just ν and ρ). To avoid this extra dependence, we chose to define d as *a* near-optimality dimension, rather than *the* near optimality dimension.

We can notice that a function with smoothness parameters (ν, ρ) has necessarily an associated constant

$$C \geq C_{\min} \triangleq \left(\frac{K}{\rho^{-d}}\right)^{\left\lfloor \frac{\log\ 3}{\log\ \frac{1}{\rho}} \right\rfloor}$$

since all the cells at depth $h_0 \triangleq \lfloor (\log 3)/(\log 1/\rho) \rfloor$ are near-optimal because of Assumption 1. Indeed, it guarantees that $\forall x \in X, f(x) \geq f(x^\star) - \nu$, which implies that $\forall x \in X, f(x) \geq f(x^\star) - 3\nu\rho^{h_0}$

We also have that $d_{\max} \triangleq (\log\ K)/(\log 1/\rho)$ is always a near-optimality dimension of the function, because of the bound K^h on the number of cell of depth h. This emphasizes the fact that the near-optimality dimension of a function is a way to characterize the complexity of optimizing the function, and not an assumption. The case $d = 0$ allows for faster rates the best empirical results. As explained by Munos (2014), the case $d = 0$ is the most relevant in practice and covers most of the real-world setups.

We now state three new different assumptions on the rate at which the cost-to-bias function Φ is decreasing, namely polynomially, exponentially, or by a constant.

Assumption 2 (Assumption on the fidelities)**.**

(a) There exist $A, \alpha > 0$ such that $\Phi(c) \leq A/c^\alpha$.

(b) There exist $B, \sigma, \beta > 0$ such that $\Phi(c) \leq Be^{\frac{-c^\beta}{\sigma}}$.

(c) There exists $a \geq 1$ such that $\Phi(c) = 0$ for all $c \geq a$.

Definition 2. *For* Asm *being one of the three Assumptions 2 (either Assumption 2(a), Assumption 2(b), Assumption 2(c), with its specific parameters depending on the case), we define $F(\text{Asm}, f, \lambda)$=$\{(f_z)_{z\in Z}|$ there exists a function ζ such that assumption* Asm *holds on f and $(f_z)_{z \in Z}$, with λ as a cost function and ζ as a bias function$\}$.*

The above assumptions describe realistic rates for the cost-to-bias function. Assumption 2(a) generalizes Assumption 3 of Sen et al. (2018) which is equivalent to the case $\alpha \geq 1$. Assumption 2(b) generalizes Assumption 2 of Sen et al. (2018) which corresponds to the case $\beta = 1$. Assumption 2(c) is relevant when a minimal cost to get a perfectly accurate estimation is needed but unknown. It is also useful to link our results (especially the theorem below) to works using single-fidelity optimization, since the settings are then equivalent to deterministic single-fidelity settings.

4 Lower bound

We provide the first lower bounds for the assumptions on the fidelities considered. Theorem 1 gives, for assumptions 2(a), 2(b) and 2(c), bounds on the achievable theoretical performance of an algorithm working under these assumptions.

Theorem 1 (Lower bounds on simple regret)**.** *Let $\mathcal{P}$ a partitioning of a space $\mathcal{X}$, (ν, ρ) some smoothness parameters, $d \in [0, d_{\max}]$ a near-optimality dimension with associated constant $C \geq C_{\min}$ and* Asm *one of the three Assumptions 2 with associated parameters. Then, for any budget Λ large enough, for any (deterministic or random) policy π, there exist a target function $f \in S(\mathcal{P}, \nu, \rho, d, C)$, a cost function λ, and fidelity approximations $(f_z)_{z\in Z} \in F(\text{Asm}, f, \lambda)$ such that:*
Under Assumption 2(a) ($\Phi(c) \leq A/c^\alpha$):

$$r_\Lambda(\pi) \geq D_1\ \Lambda^{\frac{-1}{d+\frac{1}{\alpha}}}$$

Under Assumption 2(b) ($\Phi(c) \leq Be^{\frac{-c^\beta}{\sigma}}$):

$$r_\Lambda(\pi) \geq \left| \begin{array}{ll} e^{-D_2 \Lambda^{\frac{\beta}{1+\beta}}}, & \textit{when} \quad d = 0 \\ D_3\ \Lambda^{\frac{-1}{d}}, & \textit{when} \quad d > 0 \end{array} \right.$$

Under Assumption 2(c) ($\Phi(c) = 0$ for all $c \geq a$):

$$r_\Lambda(\pi) \geq \left| \begin{array}{l} e^{-D_4 \Lambda}, \text{ when d=0} \\ D_5\, \Lambda^{\frac{-1}{d}}, \text{ when } d > 0, \end{array} \right.$$

where $D_1, D_2, D_3, D_4, D_5 > 0$ are constants that do not depend on Λ and π.

Ideas of the proof The proof is in the appendix. It is based on the construction of a target function and its approximations, such that the algorithm π may not reach a certain depth h and open a near-optimal cell at depth h. The construction of the target function is done thanks to a tree, whose leaves are cells of the partitioning, and which reflects which cells are near-optimal for the target function. The approximations are made such that we can lower bound the cost that π has to invest to get precise enough information.

We thus have to construct this tree, which is the tricky part of the proof. This implies choosing near-optimal cells that π is unlikely to open. We then get that this depth h, which depends on the parameters of the problem and on the budget, may not be reached by π with a certain fixed probability. We can use this to lower bound the regret.

Link with the upper bounds In Section 5 we give an algorithm that, without any knowledge on ν, ρ, d, C, and Φ, achieves these rates with additional constants and logarithmic factors. This means that these lower bounds are close to the optimal rate for policies working with these assumptions, both with and without knowledge of these parameters.

The only previous work using hierarchical partitioning optimization with multi-fidelity model and deterministic feedback worked with narrower assumptions as said above. It obtained, under Assumption 2(a) with $\alpha \geq 1$, a regret of $\mathcal{O}((\frac{\Lambda}{\log \Lambda})^{\frac{-1}{d+1}})$, which is only optimal (ignoring constant and log factors) when α=1. Under Assumption 2(b), MPFDOO gets, assuming $\beta = 1$ a regret of $\mathcal{O}((\frac{\Lambda}{\log \Lambda})^{\frac{-1}{d+\varepsilon}})$, for any $\varepsilon > 0$, provided the budget is large enough (with the threshold having a dependence on ε), which does not show that this lower bound was reached.

Assumption 2(c) let us extend our results to single-fidelity algorithm with deterministic feedback. A true exponential decay for $d = 0$ (and thus optimal up to a constant) was first achieved by DOO (Munos, 2011, but required the knowledge of the smoothness. SequOOL (Bartlett et al., 2019) then managed to achieve an exponential decay without the knowledge of the smoothness, but with a logarithmic factor in the exponent. We however realized it is possible to get a true exponential decay without the knowledge of the smoothness parameters by changing the number of opened cells at each depth h of SequOOL, to either $\left\lfloor 2\sqrt{n/h} \right\rfloor$ up to depth n, or $\lfloor n/\big(h \log(n/h)^2\big) \rfloor$ up to depth $\lfloor n/e^2 \rfloor$.

5 Algorithm

In this section we propose a new algorithm for multi-fidelity optimization called Kometo. We start with some helpful notation.

Cell evaluations: Cell evaluations are done through a single representant of each cell $\mathcal{P}_{h,i}$, denoted $x_{h,i}$. $T_{h,i,j}$ denote the number of evaluation potentially done for the cell $\mathcal{P}_{h,i}$ at fidelity level j.

For Kometo, the fidelity level j, with j a non-negative integer, is defined as z_{e^j}. At each fidelity level, at most one evaluation can be done for each cell, which means that $T_{h,i,j}$ is equal to either 0 or 1. We hence denote as $f_{h,i,j}$ the result of the potential evaluation, when $T_{h,i,j} = 1$. We can notice that, for any j, because of how the cells are opened, $\{\mathcal{P}_{h,i},\, T_{h,i,j}{=}1\}$ is always a tree.

We also slightly modify the usual definition of a cell opening to make it work with our multi-fidelity settings.

Multi-Fidelity Cell Opening: Opening a cell at fidelity level j means that, for each of its children $\mathcal{P}_{h+1,i}$, the $T_{h+1,i,u}$ for $0 \leq u \leq j$ are set to 1.

This means that the values $f_{h+1,i,u}$, equal to $f_{z_{e^u}}(x_{h+1,i})$, with $x_{h,i}$ the representative element of the cell, can be requested and hence the evaluations can be performed. With this definition, the opening at fidelity level j of a cell can not induce a total cost of more that $\frac{Ke^{j+1}}{e-1}$.

Kometo explanations: Kometo is detailed in Algorithm 1. The algorithm presented is inspired by StroquOOL (Bartlett et al. (2019)). Its main feature is that, using Zipf sampling (which means, opening up to $\widetilde{\Lambda}$ cells at h=1, up to $\widetilde{\Lambda}/2$ cells at h=2 and so on) it manages to reach the optimal rate up to logarithmic factor without the knowledge of the smoothness. This is done, in the exploration part, by opening a decreasing number of cell at each depth, and at a given depth, gradually decreasing the fidelity at which cells are opened. The intuition behind this idea is that, for each depth h, and each $0 \leq j_h \leq j_{\max}$, the number of cell opened at fidelity level j_h or higher will decrease with j_h. If this j_h is too low, the precision might also be too low for the choices to be relevant, but if j_h is too high, not enough cells will be opened. Cross-validation is then used by the algorithm in order

Algorithm 1 KOMETO

```
Parameters: $(f_z)_{z\in Z}$, $\mathcal{P}$, $\Lambda$, $\lambda$
Init:
$\widetilde{\Lambda} \leftarrow \frac{(e-1)\Lambda}{2Ke(\log\ \Lambda+1)^2}$, $j_{\max} \leftarrow \left\lfloor \log\ \widetilde{\Lambda} \right\rfloor$.
Open with budget $\widetilde{\Lambda}$ the cell $P_{0,0}$.

For $h = 1$ to $\lfloor\widetilde{\Lambda}\rfloor$                ◀ Exploration ▶
  For $m = 1$ to $\left\lfloor \widetilde{\Lambda}/h \right\rfloor$
    $j \leftarrow \lfloor \log\ \frac{\widetilde{\Lambda}}{hm} \rfloor$
    Open at fidelity level $j$ the non-opened

    cell $P_{h,i}$ with the highest value $f_{h,i,j}$,

    given that $T_{h,i,j} = 1$

For $j = 0$ to $j_{\max}$             ◀ Cross-validation ▶
  Evaluate at cost $\widetilde{\Lambda}$ the candidates

$x_j^c \leftarrow \underset{(h,i)\in\mathcal{T},\, T_{h,i,j}=1}{\arg\max} f_{h,i,j}$.
Output $x_\Lambda \leftarrow \underset{\{j\in[0:j_{\max}]\}}{\arg\max} f_{z_{\widetilde{\Lambda}}}\left(x_j^c\right)$
```

to choose the best cell regardless of depth and fidelity level. It ensures that the choice of a particular j_h is not needed.

Budget optimization: With a given budget Λ, we can actually initialize the $\widetilde{\Lambda}$ constant with a way higher value than $\frac{(e-1)\Lambda}{2Ke\overline{\log}\ (\Lambda+1)^2}$, for multiple reasons:

- The actual cost used for a cell opening is rounded down to $e^{\lfloor \log c \rfloor}$.
- The total budget mentioned for a cell opening assumes that all the evaluation at different fidelities will be requested for the children, which is not the case.
- The number of opened cells at each depth is bounded by K^h
- For some partitioning, it is possible to use the parent evaluations for one of its children.

Since the budget used can be predicted using only $\widetilde{\Lambda}$ and the partitioning, and increase with $\widetilde{\Lambda}$, it is possible to quickly calculate the optimal initial value of $\widetilde{\Lambda}$ using dichotomy. However, these previous optimization can only increase $\widetilde{\Lambda}$ by a multiplicative constant. Even if the budget needs to be set in advance for this algorithm, since we optimize the simple regret, we can obtain anytime guarantees which only differ by a multiplicative constant using the doubling trick.

6 Theoretical guarantees

We first state a simple proposition which asserts, with the initial value of $\widetilde{\Lambda}$, the budget condition is respected.

Proposition 2 (Budget use)**.** *The budget used by* KOMETO *does not exceed* Λ.

Our upper bounds use the Lambert function, evaluated at positive real values. This function is defined as the inverse of the function $f(z) = ze^z$. With the first two terms of its asymptotic expansion, we get, when z goes to infinity, that $W(x) = \log x - \log\log x + o(1)$.

We now state the main results of our analysis, using the same context as Theorem 1 on lower bounds. The proof is given in appendix:

Theorem 3 (Upper bounds on the simple regret)**.** *Let $\mathcal{P}$ be a partitioning of a space $\mathcal{X}$, (ν, ρ) some smoothness parameters, $d \in [0, d_{\max}]$ a near-optimality dimension with associated constant $C \geq C_{\min}$ and* Asm *being one of the three Assumption 2 with its associated parameters.*
Then, for any budget $\Lambda \geq 1$, target function $f \in S(\mathcal{P}, \nu, \rho, d, C)$, cost function λ, and fidelity approximations $(f_z)_{z\in Z} \in F(\mathrm{Asm}, f, \lambda)$ provided to KOMETO*,*

Under Assumption 2(a) $(\Phi(c) \leq \frac{A}{c^\alpha})$: We first define two values, then state the regret

Value of h_1
$\frac{1}{(d+\frac{1}{\alpha})\log\frac{1}{\rho}} W\left(\frac{\widetilde{\Lambda}\nu^{\frac{1}{\alpha}}(d+\frac{1}{\alpha})\log\frac{1}{\rho}}{4CeA^{\frac{1}{\alpha}}}\right)$

Value of h_2	
When $d = 0$	$\frac{\widetilde{\Lambda}}{4C}$
When $d > 0$	$\frac{1}{d\log\frac{1}{\rho}} W\left(\frac{\widetilde{\Lambda} d\log\frac{1}{\rho}}{4C}\right)$

Regret	
High budget *$(\nu\rho^{h_1} \leq e^\alpha A)$*	$r_\Lambda \leq \frac{3\nu}{\rho}\rho^{h_1} + 2\frac{A}{\widetilde{\Lambda}^\alpha}$
Low budget *$(\nu\rho^{h_1} > e^\alpha A)$*	$r_\Lambda \leq \frac{3\nu}{\rho}\rho^{h_2} + 2\frac{A}{\widetilde{\Lambda}^\alpha}$

Under Assumption 2(b) $(\Phi(c) \leq Be^{\frac{-c^\beta}{\sigma}})$: We also define $a_{b,\nu} = \max\left(\frac{1}{2\sigma}, log\left(\frac{B}{\nu}\right)\right)$

Value of h_1	
When $d=0$	$\left(\frac{\widetilde{\Lambda}}{4Che}\right)^{\frac{\beta}{\beta+1}}\left(\frac{1}{2\sigma\log\frac{1}{\rho}}\right)^{\frac{1}{\beta+1}}$
When $d>0$	$\frac{\beta+1}{\beta d\log\frac{1}{\rho}}W(\frac{\beta}{\beta+1}d\log\frac{1}{\rho}\left(\frac{\widetilde{\Lambda}}{4Che}\right)^{\frac{\beta}{\beta+1}}\left(\frac{1}{2\sigma\log\frac{1}{\rho}}\right)^{\frac{1}{\beta+1}})$

Value of h_2	
When $d=0$	$\frac{\widetilde{\Lambda}}{4Ce(2\sigma a_{b,\nu})^{\frac{1}{\beta}}}$
When $d>0$	$\frac{1}{d\log\frac{1}{\rho}}W\left(\frac{\widetilde{\Lambda}d\log\frac{1}{\rho}}{4Ce(2\sigma a_{b,\nu})^{\frac{1}{\beta}}}\right)$

Regret	
High budget *($h_1 \geq \frac{a_{b,\nu}}{\log\frac{1}{\rho}}$)*	$r_\Lambda \leq \frac{3\nu}{\rho}\rho^{h_1}+2Be^{\frac{-\widetilde{\Lambda}^\beta}{\sigma}}$
Low budget *($h_1 < \frac{a_{b,\nu}}{\log\frac{1}{\rho}}$)*	$r_\Lambda \leq \frac{3\nu}{\rho}\rho^{h_2}+2Be^{\frac{-\widetilde{\Lambda}^\beta}{\sigma}}$

Under Assumption 2(c) ($\Phi(c)=0$ for all $c\geq a$):

Value of h	
When $d=0$	$\frac{\widetilde{\Lambda}}{4Cae}$
When $d>0$	$\frac{1}{d\log\frac{1}{\rho}}W\left(\frac{\widetilde{\Lambda}d\log\frac{1}{\rho}}{4Cae}\right)$

Regret
$r_\Lambda \leq \frac{\nu}{\rho}\rho^h$

Corollary 4 (Regret decreasing rates)**.** *Following Theorem 3 (the exact upper bounds used for the rates are given in appendix):*

Assumption 2(a)	*High budget*	*Low budget*
When $d=0$	$\widetilde{\mathcal{O}}(\Lambda^{-\alpha})$	
When $d>0$	$\widetilde{\mathcal{O}}(\Lambda^{\frac{-1}{d+1/\alpha}})$	$\widetilde{\mathcal{O}}(\Lambda^{\frac{-1}{d}}+\Lambda^{-\alpha})$

Assumption 2(b)	*High budget*	*Low budget*
When $d=0$	$e^{\widetilde{\mathcal{O}}(-\Lambda^{\frac{\beta}{1+\beta}})}$	$e^{\widetilde{\mathcal{O}}(-\Lambda^{\beta})}+e^{\widetilde{\mathcal{O}}(-\Lambda)}$
When $d>0$	$\widetilde{\mathcal{O}}(\Lambda^{\frac{-1}{d}})$	

Assumption 2(c)	
When $d=0$	$e^{\widetilde{\mathcal{O}}(-\Lambda)}$
When $d>0$	$\widetilde{\mathcal{O}}(\Lambda^{\frac{-1}{d}})$

As explained in the next paragraph, in practice and for asymptotic comparisons only the results for high budget settings are relevant.

We can notice that the rates of decreasing are better until the threshold for high budget. This is because, until the threshold, the algorithm does not have to focus on increasing the fidelity cost to improve the result, since the improvements in the regret it can make by exploring more cells is vastly superior to the improvements it can make with more precise analysis (which involves more precise evaluations: a higher fidelity). This explains why the rates are close to the one obtained on single-fidelity optimisation (or, similarly, on Assumption 2(c), which materializes this case). However, the low budget case actually requires a very low budget (or very accurate fidelities) so these rates are not really relevant in practice. This dichotomy was similarly noticed, by Bartlett et al. (2019) for the StroquOOL algorithm, in a stochastic case: using only one evaluation was enough as long as the noise did not exceed the potential regret that could be obtained.

7 Empirical results

We chose to do the same synthetic and practical deterministic experiments that the one done in Sen et al. (2018), and used their code for fair comparisons. The algorithm to which Kometo is compared are MFPDOO (Sen et al. (2018)), POO (Grill et al. (2015)) and SequOOL (Bartlett et al. (2019)). We directly used Kometo without any tweaking. This shows Kometo adaptability, which only needed the cost function and the space $\mathcal{X}$ in order to work.

Experiments explanation Five of them are synthetic deterministic experiments of different, but always low, dimensions. The budget is expressed in terms of the number of multiple of the highest fidelity cost $\lambda(1)$. Note that these experiences may easily be unfair toward non multi-fidelity algorithms, because the results of the multi-fidelity algorithms heavily depends on how useful the low fidelities are, which is arbitrary on synthetic experiments. Therefore, since non multi-fidelity algorithms have no access to low fidelities and thus have less information, synthetic experiences should not be used to directly compare the efficiency of a multi-fidelity and a non multi-fidelity algorithm.

The last experience aims to measure the efficiency of the algorithms in practical settings. It involves tuning two hyperparameters for text classification, with the number of samples used to obtain 5-fold cross-validation accuracy determined by the fidelity. The budget is, for this experience, determined by the time used by the algorithm to return its result, reflecting simultaneously the actual time used for the algorithm execution and the cost of computing the accuracies.

Details about the experiments, along with comparisons to other multi-fidelity algorithms, can be found in Sen et al. (2018).

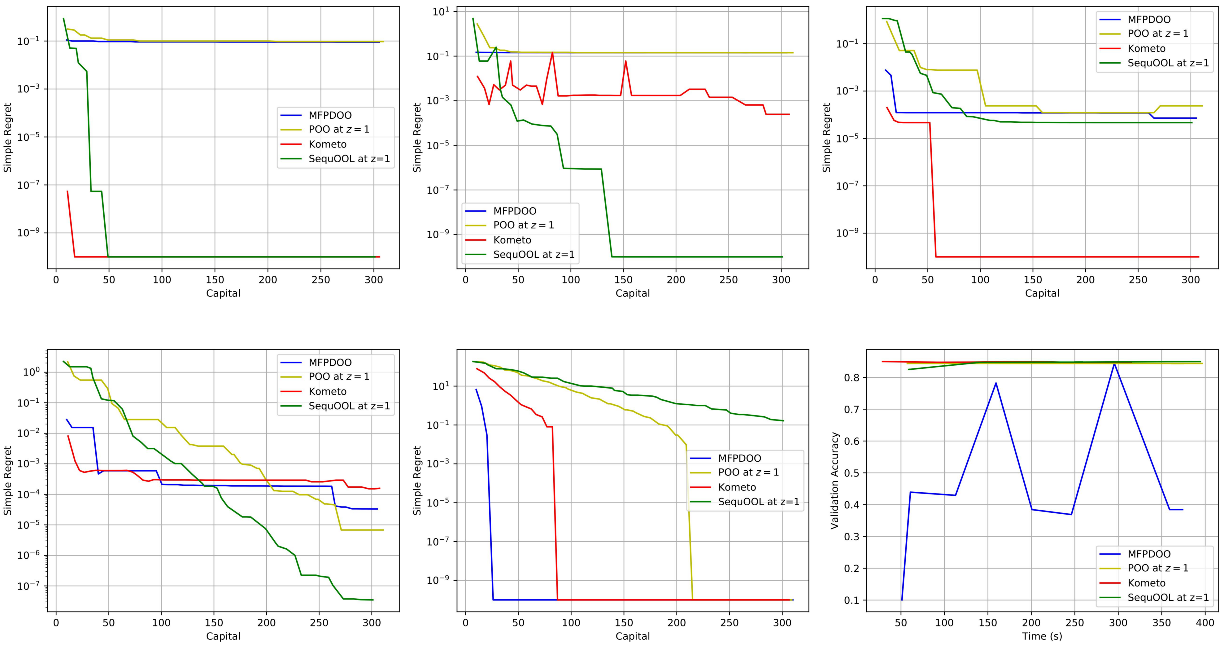

Figure 1: **(a) top-left**: Curin 2-dimensions, **(b) top-right:** Branin 2-dimensions **(c) top-left:** Hartman3d 3-dimensions **(d) bottom-right:** Hartman6d 6-dimensions **(e) bottom-right:** Borehole 8-dimensions **(f) bottom-right:** SVM 2-dimensions. Experiments are composed of five synthetic experiments, from (a) to (e), and one real-world, (f). The multi-fidelity algorithms can use all fidelities, while the non multi-fidelity algorithms only request at fidelity $z = 1$. The x-axis gives the budget effectively used by the algorithm, to reflect algorithm which exceed the attributed budget. The y-axis denotes the regret for the synthetic experiments (the lower the better), and the accuracy for the SVM experiment (the higher the better). For readability, the graph only plot the regret down to 10^{-10}.

Experiments analysis We can notice that Kometo largely outperforms MFPDOO on three of the synthetic experiments (Branin, Curin and Hartman3d) and on the practical experiment. It however gets beaten by MFPDOO on Borehole and Hartman6d by a relatively small margin. For the Hartman3d and Curin experiments, the better results of Kometo could be explained with its rank-based property, low fidelities may give highly accurate information on the way close points compare each other on the target function.

Interestingly, SequOOL outperforms Kometo on the Branin and Hartman6d experiments. This happens because, for these experiments, a lot of high-fidelity evaluations are needed to minimize the regret. Since Kometo keeps an important portion of its budget for low-fidelity evaluations, it is late compared to SequOOL which only does high-fidelity evaluations. This is materialized in the theoretical guarantees by the fact that Kometo has additionnal logarithmic factors compared to SequOOL under Assumption 2(c).

8 Discussion

Possible stochastic settings Our algorithm works in deterministic settings. However, our hypothesis of a bounded bias can be replaced with an hypothesis of a noise (potentially biased), with the same bounds. Our algorithm can therefore work in stochastic settings, the guarantees being given instead at high probability with a cost-to-bias function changed accordingly.

However, in cases where the noise does not naturally decrease to 0 at higher fidelities, the Φ function will not decrease to 0 either although required by 2. This issue can be resolved by gradually increasing the number of evaluations at higher fidelities, to get a Φ function that would converges to 0. Indeed using concentration inequalities, we could then have Assumption 2 true with high probability, which could bound the regret.

Cumulative regret in adaptive multi-fidelity optimization Locatelli and Carpentier (2018) states that the minimax optimal cumulative regret with the knowledge of the smoothness cannot be attained by single-fidelity algorithm without the knowledge of the smoothness of the function. We wonder if this result remains true in multi-fidelity settings using adapted cumulative regret definitions.

References

Auer, P., Ortner, R., and Szepesvári, C. (2007). Improved rates for the stochastic continuum-armed bandit problem. In *Conference on Learning Theory*.

Bartlett, P. L., Gabillon, V., and Valko, M. (2019). A simple parameter-free and adaptive approach to optimization under a minimal local smoothness assumption. In *Algorithmic Learning Theory*.

Bubeck, S., Munos, R., Stoltz, G., and Szepesvári, C. (2011). X-armed bandits. *Journal of Machine Learning Research*, 12:1587–1627.

Cutler, M., Walsh, T. J., and How, J. P. (2014). Reinforcement learning with multi-fidelity simulators. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3888–3895. IEEE.

Ghosh, S., Kristensen, J., Zhang, Y., Subber, W., and Wang, L. (2019). A Strategy for Adaptive Sampling of Multi-fidelity Gaussian Process to Reduce Predictive Uncertainty. *preprint*.

Grill, J.-B., Valko, M., and Munos, R. (2015). Blackbox optimization of noisy functions with unknown smoothness. In *Neural Information Processing Systems*.

Hoorfar, A. and Hassani, M. (2008). Inequalities on the Lambert W function and hyperpower function. *Journal of Inequalities in Pure and Applied Mathematics*, 9(2):5–9.

Huang, D., Allen, T. T., Notz, W. I., and Miller, R. A. (2006). Sequential kriging optimization using multiple-fidelity evaluations. *Structural and Multidisciplinary Optimization*, 32(5):369–382.

Kandasamy, K., Dasarathy, G., Oliva, J. B., Schneider, J., and Poczos, B. (2016a). Gaussian process bandit optimisation with multi-fidelity evaluations. *30th Conference on Neural Information Processing Systems*.

Kandasamy, K., Dasarathy, G., Oliva, J. B., Schneider, J., and Poczos, B. (2016b). Multi-fidelity Gaussian Process Bandit Optimisation. *Journal of Artificial Intelligence Research*.

Kandasamy, K., Dasarathy, G., Poczos, B., and Schneider, J. (2016c). The multi-fidelity multi-armed bandit. In *Advances in Neural Information Processing Systems*, pages 1777–1785.

Kandasamy, K., Dasarathy, G., Schneider, J., and Poczos, B. (2017). Multi-fidelity bayesian optimisation with continuous approximations. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1799–1808. JMLR. org.

Kawaguchi, K., Maruyama, Y., and Zheng, X. (2016). Global continuous optimization with error bound and fast convergence. *Journal of Artificial Intelligence Research*, 56:153–195.

Kleinberg, R., Slivkins, A., and Upfal, E. (2008). Multiarmed bandit problems in metric spaces. In *Symposium on Theory Of Computing*.

Li, L., Jamieson, K., DeSalvo, G., and Talwalkar, A. R. A. (2017). Hyperband: Bandit-based configuration evaluation for hyperparameter optimization. In *International Conference on Learning Representations*.

Locatelli, A. and Carpentier, A. (2018). Adaptivity to Smoothness in X-armed bandits. In *Conference on Learning Theory*.

Matyas, J. (1965). Random optimization. *Automation and Remote control*, 26(2):246–253.

Munos, R. (2011). Optimistic optimization of deterministic functions without the knowledge of its smoothness. In *Neural Information Processing Systems*.

Munos, R. (2014). From Bandits to Monte-Carlo Tree Search: The Optimistic Principle Applied to Optimization and Planning. *Foundations and Trends in Machine Learning*.

Nesterov, Y. and Spokoiny, V. (2017). Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17(2):527–566.

Sen, R., Kandasamy, K., and Shakkottai, S. (2018). Multi-Fidelity Black-Box Optimization with Hierarchical Partitions. *International Conference on Machine Learning*.

Sen, R., Kandasamy, K., and Shakkottai, S. (2019). Noisy Blackbox Optimization with Multi-Fidelity Queries: A Tree Search Approach. *Proceedings of the 22nd International Conference on Artificial Intelligence and Statistics*.

Shang, X., Kaufmann, E., and Valko, M. (2019). General parallel optimization without metric. In *Algorithmic Learning Theory*.

Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M. (2010). Gaussian process optimization in the bandit setting: No regret and experimental design. *International Conference on Machine Learning*.

Valko, M., Carpentier, A., and Munos, R. (2013). Stochastic simultaneous optimistic optimization. In *International Conference on Machine Learning*.

Zhang, Y., Hoang, T. N., Kian, B., Low, H., and Kankanhalli, M. (2019). Information-Based Multi-Fidelity Bayesian Optimization. *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*.