Relaxing the Additivity Constraints in Decentralized No-Regret High-Dimensional Bayesian Optimization

Anthony Bardou
ENS Lyon, UCBL, CNRS, LIP
Lyon, France
anthony.bardou@ens-lyon.fr

Patrick Thiran
IC, EPFL
Lausanne, Switzerland
patrick.thiran@epfl.ch

Thomas Begin
ENS Lyon, UCBL, CNRS, LIP
Lyon, France
thomas.begin@ens-lyon.fr

Abstract

Bayesian Optimization (BO) is typically used to optimize an unknown function f that is noisy and costly to evaluate, by exploiting an acquisition function that must be maximized at each optimization step. Although provably asymptotically optimal BO algorithms are efficient at optimizing low-dimensional functions, scaling them to high-dimensional spaces remains an open research problem, often tackled by assuming an additive structure for f. However, such algorithms introduce additional restrictive assumptions on the additive structure that reduce their applicability domain. In this paper, we relax the restrictive assumptions on the additive structure of f, at the expense of weakening the maximization guarantees of the acquisition function, and we address the over-exploration problem for decentralized BO algorithms. To these ends, we propose DumBO, an asymptotically optimal decentralized BO algorithm that achieves very competitive performance against state-of-the-art BO algorithms, especially when the additive structure of f does not exist or comprises high-dimensional factors.

1 Introduction

Many real-world applications involve the optimization of an unknown objective function f that is noisy and costly to evaluate. Examples of such tasks include hyper parameters tuning in deep neural networks [1], robotics [2], networking [3] and computational biology [4]. In such applications, f can be seen as a black box that can only be discovered by successive queries. This prevents the use of traditional first order approaches to optimize f.

Bayesian Optimization (BO) has become a highly effective framework for black-box optimization. Typically, a BO algorithm tackles this problem by modeling f as a Gaussian process (GP) and by leveraging this model to query f at specific inputs. The challenge of querying f is to trade off exploration (*i.e.* to query an input that improves the quality of the GP regression of f) for exploitation (*i.e.* to query an input that is thought to be the maximal argument of f). To achieve this trade-off at time t, a BO algorithm maximizes an acquisition function $\varphi_t(\boldsymbol{x})$, built by leveraging the information provided by the GP model, to select a query $\boldsymbol{x}^t$.

Although BO has shown its efficiency at optimizing black-box functions, so far it has mostly found success with low dimensional input spaces [5]. However real-world applications, such as computer vision, robotics or networking, often involve a high-dimensional objective function f.

Preprint.

Scaling classical BO algorithms to such input spaces remains a great challenge as the cost of finding $\arg\max \varphi_t$ grows exponentially with the input space dimension d. A way to circumvent that issue is to cap the complexity of the maximization by assuming an additive decomposition of f [6–8] with a low *Maximum Factor Size* (MFS), denoted $\bar{d}$ and corresponding to the maximum number of dimensions for a factor of the decomposition. Unfortunately, assuming an additive decomposition with low MFS may lead to the optimization of a coarse approximation of f. Since the low MFS assumption is only needed to ensure that the global maximum of φ_t is found within a reasonable amount of time, a methodological question arises: is it better (i) to have a guarantee of reaching the global maximum on an acquisition function built from a simple (and often inexact) decomposition of f, or (ii) to use an acquisition function built from a more complex but exact decomposition of f by giving up on the guarantee of reaching the global maximum?

Case (i) has been extensively studied [6–8], but it seems that only [7] has taken a few steps in the direction of case (ii). This article embraces case (ii) and demonstrates that it is possible to relax the low-MFS assumptions that limit the applicability domain of asymptotically optimal BO algorithms. To illustrate case (ii), we propose DuMBO, a decentralized, message-passing, provably asymptotically optimal BO algorithm able to infer a complex additive decomposition of f without any assumption regarding its MFS. Additionally, we provide a more efficient way to approximate the well-known GP-UCB acquisition function [9] in a decentralized context. Finally, we evaluate the pros and cons of case (ii) by comparing DuMBO with several state-of-the-art solutions on both synthetic and real-world problems wherein the noisy objective function f cannot be decomposed into low-dimensional factors.

2 Background

2.1 State of the Art

Given a black-box, costly to evaluate objective function $f : \mathcal{D} \subset \mathbb{R}^d \to \mathbb{R}$, the goal of a BO algorithm is to find $\arg\max_{\boldsymbol{x}} f(\boldsymbol{x})$ using as few queries as possible. To quantify the quality of the optimization, one can consider the immediate regret $r_t = f(\boldsymbol{x}^*) - f(\boldsymbol{x}^t)$ (note that $\boldsymbol{x}^* = \arg\max_{\boldsymbol{x}} f(\boldsymbol{x})$) and attempt to minimize the cumulative regret $R_t = \sum_{i=1}^{t} r_i$. A BO algorithm is said to be asymptotically optimal if $\lim_{t\to+\infty} R_t/t = 0$, which implies that the BO algorithm will asymptotically reach $\boldsymbol{x}^*$ and hence guarantees *no-regret* performance.

A BO algorithm typically uses a GP to infer a posterior distribution for the value of $f(\boldsymbol{x})$ at any point $\boldsymbol{x} \in \mathcal{D}$ and selects, at each time step t, a query $\boldsymbol{x}^t$. The BO algorithm bases its querying policy on the maximization of an acquisition function that quantifies the benefits of observing $f(\boldsymbol{x})$ in terms of exploration and exploitation. Common acquisition functions include probability of improvement [10], expected improvement [11] and upper confidence bound [12]. The latter leads to an asymptotically optimal application to GPs, with GP-UCB [9], defined as

$$\varphi_t(\boldsymbol{x}) = \mu_t(\boldsymbol{x}) + \beta_t^{\frac{1}{2}} \sigma_t(\boldsymbol{x}). \tag{1}$$

It involves an exploitation term $\mu_t(\boldsymbol{x})$, which is the posterior mean of the GP at input $\boldsymbol{x}$, and an exploration term $\sigma_t(\boldsymbol{x})$, which is the posterior standard deviation of the GP at input $\boldsymbol{x}$. Finally, the scalar $\beta_t^{1/2}$ handles the exploration-exploitation trade-off in order to guarantee the asymptotic optimality of GP-UCB with high probability.

As stated before, scaling BO algorithms to high-dimensional functions is challenging because of the exponential complexity of the global optimization algorithms used to maximize the acquisition function φ_t. To tackle this problem, BO algorithms generally fall into one of these two categories (with the exception of TuRBO [13], which uses trust regions to maximize f).

Embedding BO algorithms assume that only a few dimensions significantly impact f and project the high-dimensional space of f into a low-dimensional one where the optimization is actually performed. REMBO [14] and ALEBO [15] use random matrices to embed the high-dimensional space while SAASBO [16] uses sparse GPs defined on subspaces. Other approaches [17, 18] are based, respectively, on Variational Auto-Encoders and on manifold GPs to learn an embedding. Finally, there exists approaches that select some dimensions of the input space to project onto. Such recent methods include Dropout [19] and MCTS-VS [20], based on Monte-Carlo Tree Search.

Table 1: Comparison of decomposing state-of-the-art BO algorithms with DuMBO on relevant criteria. Here, n is the number of factors in the decomposition, d the number of dimensions of f, $\bar{d}$ the MFS of the decomposition, t the optimization step, ζ the desired accuracy when maximizing φ_t and N_A a constant defined in Appendix D. N_m is a constant defined in [7].

Solution	Complexity	MFS Assumption	Find $\arg\max \varphi_t$
ADD-GPUCB	$\mathcal{O}\left(t^3 + nt^2 + n^2\zeta^{-1}\right)$	$\bar{d} = 1$	Yes
QFF	$\mathcal{O}\left((\zeta^{-1}t^{3/2}(\log t)^{\bar{d}})^{\bar{d}}\right)$	$\bar{d} = 1$	Yes
DEC-HBO	$\mathcal{O}\left(N_m\zeta^{-\bar{d}}n(t^3 + n)\right)$	$\bar{d} \leq 3$	If the decomposition is sparse
DuMBO	$\mathcal{O}\left(\bar{d}N_A n t^3 \zeta^{-1}\right)$	None	If φ_t is restricted prox-regular

Decomposing BO algorithms assume an additive structure for f and optimize the factors of the induced decomposition. Classical approaches such as MES [21], ADD-GPUCB [6] or QFF [8] assume a decomposition with a MFS equal to 1 and orthogonal domains. More recent approaches like DEC-HBO [7] are able to optimize decompositions with larger MFS and shared input components. Still, the MFS of the decomposition must be low to avoid a prohibitive computational complexity. Note that, under some assumptions on f, these approaches are provably asymptotically optimal and a subset of them, namely ADD-GPUCB [6] and DEC-HBO [7], can be used in a decentralized context.

2.2 DuMBO (Decentralized Message-passing Bayesian Optimization algorithm)

In this article, we propose DuMBO, a decomposing algorithm that relaxes the low MFS constraint on the assumed additive decomposition of f. Table 1 gathers the main differences between DuMBO and state-of-the-art decomposing algorithms. Note that ADD-GPUCB and QFF require the simplest form of additive decompositions ($\bar{d} = 1$). As a consequence, when optimizing a complex objective function f, they need to approximate it with a decomposition with MFS $\bar{d} = 1$. In return, they are systematically able, at each time step t, to query $\arg\max \varphi_t$. Observe that DEC-HBO tolerates more complex decompositions ($\bar{d} = 3$), but is no longer guaranteed to find the global maximum of φ_t (because it uses a variant of the max-sum algorithm [22] that requires f to have a sparse additive decomposition to converge). Overall, DuMBO is the only algorithm that exploits weaker guarantees on the maximization of φ_t to lower its computational complexity. This allows DuMBO to handle decompositions with an arbitrary MFS without the need to approximate them with a simpler one.

The remaining of this article is devoted to formulating the BO problem (Section 3), presenting DuMBO and its early-stopped version (Section 4), providing theoretical guarantees (Section 5) and comparing them with state-of-the-art BO algorithms (Section 6).

3 Problem Formulation and First Results

In this section, we introduce the core assumptions about the black-box objective function $f : \mathcal{D} \to \mathbb{R}$ to obtain an additive decomposition (Section 3.1). Next, we exploit these assumptions to derive inference formulas (Section 3.2) and to adapt GP-UCB to a decentralized context (Section 3.3).

3.1 Core Assumptions

In order to optimize f in a decentralized fashion, we make several assumptions.

Assumption 3.1. The unknown objective function f can be decomposed into a sum of factor functions $\left(f^{(i)}\right)_{i\in[1,n]}$, with domains $\left(D^{(i)}\right)_{i\in[1,n]}$, such that $\mathcal{D} = \cup_{i=1}^{n} D^{(i)}$ and

$$f = \sum_{i=1}^{n} f^{(i)}. \tag{2}$$

Any decomposition, including (2), can be represented by a factor graph where each factor and variable node denote, respectively, one of the n factors of the decomposition and one of the d input components of f. An edge exists between a factor node i and a variable node j if and only if $f^{(i)}$ uses

x_j as an input component. We use $\mathcal{V}_i, 1 \leq i \leq n$, and $\mathcal{F}_j, 1 \leq j \leq d$, to denote respectively the set of variable nodes connected to factor node i and the set of factor nodes connected to variable node j.

To make predictions about the factor functions without any prior knowledge, we need a model that maps the previously collected inputs with their noisy outputs. Denoting $\boldsymbol{x}_{\mathcal{V}_i} = (x_j)_{j \in \mathcal{V}_i}$, let us introduce the following assumption.

Assumption 3.2. Factor functions $f^{(i)}$ are independent $\mathcal{GP}\left(\mu_0^{(i)}, k^{(i)}\left(\boldsymbol{x}_{\mathcal{V}_i}, \boldsymbol{x}'_{\mathcal{V}_i}\right)\right)$, with prior mean $\mu_0^{(i)} = 0$ and covariance function $k^{(i)}$.

Since f is a sum of independent GPs, Assumption 3.2 implies that f is also $\mathcal{GP}\left(\mu_0, k(\boldsymbol{x}, \boldsymbol{x}')\right)$ with prior mean $\mu_0 = 0$ and covariance function $k(\boldsymbol{x}, \boldsymbol{x}') = \sum_{i=1}^{n} k^{(i)}\left(\boldsymbol{x}_{\mathcal{V}_i}, \boldsymbol{x}'_{\mathcal{V}_i}\right)$.

3.2 Inference Formulas

For any $\boldsymbol{x} \in \mathcal{D}$ and given the previous t input queries $(\boldsymbol{x}^1, \cdots, \boldsymbol{x}^t)$, the vector $(f(\boldsymbol{x}), f(\boldsymbol{x}^1), \cdots, f(\boldsymbol{x}^t))$ is Gaussian. Given the t-dimensional vector of noisy outputs $\boldsymbol{y} = (y_1, \cdots, y_t)^\top$, with $y_i = f(\boldsymbol{x}) + \epsilon$ and ϵ a centered Gaussian variable, the posterior distribution of the factor $f^{(i)}(\boldsymbol{x})$ is also Gaussian. Since f can be decomposed, the posterior mean $\mu_{t+1}^{(i)}(\boldsymbol{x}_{\mathcal{V}_i})$ and variance $(\sigma_{t+1}^{(i)}(\boldsymbol{x}_{\mathcal{V}_i}))^2$ of the factor $f^{(i)}$ at time $t+1$ can be expressed with the posterior means and covariance functions of the factor functions involved in decomposition (2).

Proposition 3.3. *Let $\mu_{t+1}^{(i)}(\boldsymbol{x}_{\mathcal{V}_i})$ and $(\sigma_{t+1}^{(i)}(\boldsymbol{x}_{\mathcal{V}_i}))^2$ be the posterior mean and variance of $f^{(i)}$ at input $\boldsymbol{x}_{\mathcal{V}_i}$. Then, for the decomposition (2),*

$$\mu_{t+1}^{(i)}(\boldsymbol{x}_{\mathcal{V}_i}) = \boldsymbol{k}_{\boldsymbol{x}_{\mathcal{V}_i}}^{(i)\top} \boldsymbol{K}^{-1} \boldsymbol{y} \tag{3}$$

$$(\sigma_{t+1}^{(i)}(\boldsymbol{x}_{\mathcal{V}_i}))^2 = k^{(i)}(\boldsymbol{x}_{\mathcal{V}_i}, \boldsymbol{x}_{\mathcal{V}_i}) - \boldsymbol{k}_{\boldsymbol{x}_{\mathcal{V}_i}}^{(i)\top} \boldsymbol{K}^{-1} \boldsymbol{k}_{\boldsymbol{x}_{\mathcal{V}_i}}^{(i)} \tag{4}$$

with $t \times 1$ vectors $\boldsymbol{k}_{\boldsymbol{x}_{\mathcal{V}_i}}^{(i)} = (k^{(i)}(\boldsymbol{x}_{\mathcal{V}_i}, \boldsymbol{x}_{\mathcal{V}_i}^j))_{j \in [1,t]}$ and $t \times t$ matrices $\boldsymbol{K} = (k(\boldsymbol{x}_{\mathcal{V}_i}^j, \boldsymbol{x}_{\mathcal{V}_i}^k))_{j,k \in [1,t]}$.

For the sake of generality, Proposition 3.3 only requires an additive decomposition of f. Appendix A describes how such a decomposition can be inferred from data, using the method proposed by [23], similarly to Appendix B of [7]. Note that Proposition 3.3 does *not* assume a corresponding additive decomposition of the observed outputs in $\boldsymbol{y}$. However, note that, in a significant portion of real-world applications (e.g. network throughput maximization [24], energy consumption minimization [25] or UAVs-related applications [26]), a natural output decomposition is observable. As demonstrated by [27], having access to a decomposed output can only improve the predictive performance of the GP surrogate model. Therefore, we derive the inference formulas when the output decomposition is known in Appendix B. Also, we explore the impact of having access to the decomposed output of f in Section 6.

3.3 Proposed Acquisition Function

Having defined a surrogate model for f, we can now turn to finding an optimal policy for querying the objective function. In this section, we exploit the decomposition of f to build an acquisition function for our BO algorithm that approximates GP-UCB in a decentralized context. Proofs for all the presented results can be found in Appendix C.

Recall that GP-UCB is defined by (1) as the sum of an exploitation term $\mu_t(\boldsymbol{x})$ and an exploration term $\sigma_t(\boldsymbol{x})$ weighted by some scalar $\beta_t^{1/2}$. Finding an additive decomposition for GP-UCB is hard because $\sigma_t(\boldsymbol{x})$ cannot be expressed as a sum. To circumvent this caveat, [6] proposed to apply GP-UCB to each factor of the additive decomposition of f, with $\varphi_t^{(i)} = \mu_t^{(i)} + \beta_t^{1/2} \sigma_t^{(i)}$. Then, they proved that their algorithm ADD-GPUCB offers no-regret performance by considering $\sum_{i=1}^{n} \varphi_t^{(i)} = \mu_t + \beta_t^{1/2} \sum_{i=1}^{n} \sigma_t^{(i)}$ as an acquisition function. Although the exploitation term μ_t is preserved, the exploration term is now overweighted since $\sum_{i=1}^{n} \sigma_t^{(i)} \geq \sqrt{\sum_{i=1}^{n} \left(\sigma_t^{(i)}\right)^2} = \sigma_t$. To reach better empirical performance, one could look for a tighter additive upper bound of σ_t^2. This is

the purpose of this section. We start by decomposing the variance of the ith factor function $(\sigma_t^{(i)}(\boldsymbol{x}))^2$ into two terms.

Epistemic vs. aleatoric uncertainty. The variance $(\sigma_t^{(i)}(\boldsymbol{x}))^2$ of the random variable $f^{(i)}(\boldsymbol{x})$ is composed of two fundamentally different terms. The *epistemic* uncertainty refers to the uncertainty caused by having an undersampled dataset with not enough data points to accurately estimate the value of $f^{(i)}(\boldsymbol{x})$. In contrast, the *aleatoric* uncertainty can intuitively be seen as the observational noise of $f^{(i)}(\boldsymbol{x})$ due, for instance, to a poor measurement quality. For more details, refer to [28].

Let $v_-^{(i)} \geq 0$ be the aleatoric uncertainty of the factor function $f^{(i)}$, which can be seen as a lower bound of the posterior variance of $f^{(i)}$, that is $\forall \boldsymbol{x} \in \mathcal{D}^{(i)}, \forall t \in \mathbb{N}$, $v_-^{(i)} \leq (\sigma_t^{(i)}(\boldsymbol{x}))^2$. A better approximation of the exploration term can be proposed if the posterior variance of the GP is assumed to be bounded.

Assumption 3.4. $\forall t \in \mathbb{N}$, the posterior variance of the objective function f, $\sigma_t^2(\boldsymbol{x}) = \sum_{i=1}^n \left(\sigma_t^{(i)}(\boldsymbol{x}_{\mathcal{V}_i})\right)^2$ satisfies $\sigma_t^2(\boldsymbol{x}) \leq v_+$, with $v_+ = \left(\sqrt{v_-} + 2\delta_-\right)^2$, $v_- = \sum_{i=1}^n v_-^{(i)}$ and $\delta_-^2 = \sum_{i=1}^n \sum_{\substack{j=1 \\ j \neq i}}^n \sqrt{v_-^{(i)} v_-^{(j)}}$.

Note that the restrictiveness of Assumption 3.4 fades as the number n of factors grows.

Example. Consider the case where all the n factor functions $f^{(i)}$ have the same aleatoric uncertainty v_0. We have $v_- = nv_0$, $\delta_-^2 = n(n-1)v_0$ and $v_+ = nv_0(1 + 2\sqrt{n-1})^2$. Thus, the ratio $v_+/v_- = (1 + 2\sqrt{n-1})^2$ increases as n grows, which suggests that Assumption 3.4 is more easily verified when n is large. Note that v_+/v_- can be quite large for reasonable values of n. Considering a decomposition with $n = 6$, $v_+/v_- \approx 30$. In this particular context where we consider an additive decomposition of f composed of 6 factors with the same aleatoric uncertainty v_0, Assumption 3.4 holds for any objective function whose posterior variance is less than *30 times* its aleatoric uncertainty.

Under Assumption 3.4, we propose to bound from above the exploration term with the following proposition.

Proposition 3.5. *Under Assumption 3.4,*

$$a \sum_{i=1}^n {\sigma_t^{(i)}}^2 + \frac{1}{4a} \tag{5}$$

is the best linear overestimation of the exploration term σ_t (in the least squares sense), where a is the single positive real root of the quartic polynomial

$$P(a) = \frac{2\left[u^3\right]_{v_-}^{v_+}}{3} a^4 - \frac{4\left[u^{\frac{5}{2}}\right]_{v_-}^{v_+}}{5} a^3 + \frac{\left[u^{\frac{3}{2}}\right]_{v_-}^{v_+}}{3} a - \frac{[u]_{v_-}^{v_+}}{8} \tag{6}$$

and $[h(u)]_{v_-}^{v_+} = h(v_+) - h(v_-)$.

We show that (5) is a tighter upper bound of $\sigma_t(\boldsymbol{x})$ than the one proposed in [6].

Theorem 3.6. *Let Assumptions 3.1, 3.2, 3.4 hold. Then the following inequality holds for all $\boldsymbol{x} \in \mathcal{D}$*

$$a \sum_{i=1}^n \left(\sigma_t^{(i)}(\boldsymbol{x}_{\mathcal{V}_i})\right)^2 + \frac{1}{4a} \leq \sum_{i=1}^n \sigma_t^{(i)}(\boldsymbol{x}_{\mathcal{V}_i}). \tag{7}$$

Therefore, we propose an acquisition function $\varphi_t = \sum_{i=1}^n \varphi_t^{(i)}$ corresponding to the described additive decomposition (2) with

$$\varphi_t^{(i)}(\boldsymbol{x}_{\mathcal{V}_i}) = \mu_t^{(i)}(\boldsymbol{x}_{\mathcal{V}_i}) + a\beta_t^{\frac{1}{2}} \left(\sigma_t^{(i)}(\boldsymbol{x}_{\mathcal{V}_i})\right)^2 \tag{8}$$

4 Proposed Algorithm

In this section, we describe DuMBO, a BO algorithm that exploits the results from Section 3 to find $\arg\max_{\boldsymbol{x} \in \mathcal{D}} \sum_{i=1}^n \varphi_t^{(i)}(\boldsymbol{x}_{\mathcal{V}_i})$, which comes down to $\arg\max_{\boldsymbol{x} \in \mathcal{D}} \mu_t(\boldsymbol{x}) + \beta_t^{1/2} \left(a\sigma_t^2(\boldsymbol{x}) + 1/4a\right)$ since both expressions differ only by a constant term. We also provide an early-stopped version of DuMBO and discuss the weaker guarantees achieved in this case.

4.1 DuMBO

Optimizing $\varphi_t(\boldsymbol{x}) = \sum_{i=1}^{n} \varphi_t^{(i)}(\boldsymbol{x}_{\mathcal{V}_i})$ while ensuring the compatibility between shared input components amounts to solving the following constrained optimization problem:

$$\max \sum_{i=1}^{n} \varphi_t^{(i)} \left(\boldsymbol{x}^{(i)}\right) \text{ such that } \boldsymbol{x}^{(i)}_{\mathcal{V}_i \cap \mathcal{V}_j} = \boldsymbol{x}^{(j)}_{\mathcal{V}_i \cap \mathcal{V}_j}, \forall i, j \in [1, n] \tag{9}$$

with $\boldsymbol{x}^{(1)}, \cdots, \boldsymbol{x}^{(n)}$ inputs (with dimension indices respectively listed in $\mathcal{V}_1, \cdots, \mathcal{V}_n$) of the factor functions $\varphi_t^{(1)}, \cdots, \varphi_t^{(n)}$.

To simplify the expression of the equality constraints (9), we introduce a global consensus variable $\bar{\boldsymbol{x}} \in \mathcal{D}$ and we reformulate the optimization problem as

$$\max \sum_{i=1}^{n} \varphi_t^{(i)} \left(\boldsymbol{x}^{(i)}\right) \text{ such that } \boldsymbol{x}^{(i)} = \bar{\boldsymbol{x}}_{\mathcal{V}_i}, \forall i \in [1, n]\,. \tag{10}$$

We now turn the problem (10) into an unconstrained optimization problem by considering its augmented Lagrangian $\mathcal{L}_\eta(\boldsymbol{x}^{(1)}, \cdots, \boldsymbol{x}^{(n)}, \bar{\boldsymbol{x}}, \boldsymbol{\lambda})$:

$$\mathcal{L}_\eta = \sum_{i=1}^{n} \varphi_t^{(i)}(\boldsymbol{x}^{(i)}) - \boldsymbol{\lambda}^{(i)\top}(\boldsymbol{x}^{(i)} - \bar{\boldsymbol{x}}_{\mathcal{V}_i}) - \frac{\eta}{2}||\boldsymbol{x}^{(i)} - \bar{\boldsymbol{x}}_{\mathcal{V}_i}||_2^2 \tag{11}$$

with $\boldsymbol{\lambda}_k^{(i)}$ a column vector of dual variables with $|\mathcal{V}_i|$ components and a hyperparameter $\eta > 0$.

To maximize (11), we consider the Alternating Direction Method of Multipliers (ADMM), proposed by [29]. We now describe how we apply ADMM to our problem and present some relevant well-known results. For further details, please refer to [30].

ADMM is an iterative method that proposes, at iteration k, to solve sequentially the problems

$$\begin{aligned}
\boldsymbol{x}_{k+1}^{(1)} &= \arg\max_{\boldsymbol{x}^{(1)}} \mathcal{L}(\boldsymbol{x}^{(1)}, \cdots, \boldsymbol{x}_k^{(n)}, \bar{\boldsymbol{x}}_k, \boldsymbol{\lambda}_k) \\
&\vdots \\
\boldsymbol{x}_{k+1}^{(n)} &= \arg\max_{\boldsymbol{x}^{(n)}} \mathcal{L}(\boldsymbol{x}_{k+1}^{(1)}, \cdots, \boldsymbol{x}_{k+1}^{(n-1)}, \boldsymbol{x}^{(n)}, \bar{\boldsymbol{x}}_k, \boldsymbol{\lambda}_k) \\
\bar{\boldsymbol{x}}_{k+1} &= \arg\max_{\bar{\boldsymbol{x}}} \mathcal{L}(\boldsymbol{x}_{k+1}^{(1)}, \cdots, \boldsymbol{x}_{k+1}^{(n)}, \bar{\boldsymbol{x}}, \boldsymbol{\lambda}_k) && (12) \\
\boldsymbol{\lambda}_{k+1} &= \arg\max_{\boldsymbol{\lambda}} \mathcal{L}(\boldsymbol{x}_{k+1}^{(1)}, \cdots, \boldsymbol{x}_{k+1}^{(n)}, \bar{\boldsymbol{x}}_{k+1}, \boldsymbol{\lambda}). && (13)
\end{aligned}$$

Note that $\boldsymbol{x}_{k+1}^{(1)}, \cdots, \boldsymbol{x}_{k+1}^{(n)}$ can be found concurrently, by gradient ascent (e.g. with ADAM [31]) of

$$\mathcal{L}_\eta^{(i)} = \varphi_t^{(i)}(\boldsymbol{x}^{(i)}) - \boldsymbol{\lambda}^{(i)\top}(\boldsymbol{x}^{(i)} - \bar{\boldsymbol{x}}_{\mathcal{V}_i}) - \frac{\eta}{2}||\boldsymbol{x}^{(i)} - \bar{\boldsymbol{x}}_{\mathcal{V}_i}||_2^2. \tag{14}$$

If $\forall i \in [1, n]$, $\sum_{j \in \mathcal{F}_i} \lambda_{0,i}^{(j)} = 0$, it is known (see [30]) that the closed-forms for (12) and (13) are

$$\bar{\boldsymbol{x}}_{k+1} = \left(\frac{1}{|\mathcal{F}_i|} \sum_{j \in \mathcal{F}_i} x_{k+1,i}^{(j)} \right)_{i \in [1,d]} \tag{15}$$

$$\boldsymbol{\lambda}_{k+1} = \left(\boldsymbol{\lambda}_k^{(i)} + \eta \left(\boldsymbol{x}_{k+1}^{(i)} - \bar{\boldsymbol{x}}_{k+1, \mathcal{V}_i} \right) \right)_{i \in [1,n]}. \tag{16}$$

These results describe a fully decentralized message-passing algorithm, called DuMBO, that can run on the factor graph of f. A discussion about its time complexity is given in Appendix D. Since DuMBO relies on ADMM to maximize $\varphi_t = \sum_{i=1}^{n} \varphi_t^{(i)}$, let us briefly discuss its maximization guarantees. It is well known that ADMM converges towards the global maximum of a convex φ_t. ADMM has also demonstrated very good performance at optimizing non-convex functions [32–34]. This has been explained by recent works such as [35], which extends the global maximization guarantee of ADMM to the class of *restricted prox-regular* functions, which includes some non-convex, non-smooth functions.

4.2 Early-stopping

Although DuMBO has a competitive time complexity (see Table 1), we now discuss how to properly early-stop it and still get (weaker) guarantees on the maximization of φ_t *before* ADMM converges. This can be of critical importance for some real-world applications. Note that the proofs for the results in this section can be found in Appendix E. We start with the following assumption.

Assumption 4.1. The covariance functions $k^{(i)}$ from Assumption 3.2 are Lipschitz continuous, with Lipschitz constant $L\left(k^{(i)}\right) = \max_{\boldsymbol{x},\boldsymbol{x}',\boldsymbol{y}\in D^{(i)}} \frac{|k^{(i)}(\boldsymbol{x},\boldsymbol{x}')-k^{(i)}(\boldsymbol{y},\boldsymbol{x}')|}{||\boldsymbol{x}-\boldsymbol{y}||_2}$.

Assumption 4.1 holds for a large class of covariance functions, such as Matérn or the squared exponential. For such covariance functions, we have the following result.

Proposition 4.2. *Let Assumptions 3.1, 3.2, 3.4 and 4.1 hold. Then, $\varphi_t^{(i)}$ is Lipschitz continuous, with Lipschitz constant*

$$L\left(\varphi_t^{(i)}\right) = tL\left(k^{(i)}\right)\rho\left(\boldsymbol{K}^{-1}\right)M_t^{(i)} \tag{17}$$

with $M_t^{(i)} = \max\left(|y_t^+ - 2a\beta_t^{1/2}v_-^{(i)}|, |y_t^- - 2a\beta_t^{1/2}v_+^{(i)}|\right)$, $y_t^+ = \max_{k\in[1,t]} y_k$, $y_t^- = \min_{k\in[1,t]} y_k$ *and* $\rho\left(\boldsymbol{K}^{-1}\right)$ *the spectral radius of* $\boldsymbol{K}^{-1}$.

Thanks to the Lipchitz continuity of the acquisition function, we have the following result.

Theorem 4.3. *$\forall i \in [1,n]$, let $\boldsymbol{x}_{k+1}^{(i)} = \arg\max_{\boldsymbol{x}} \mathcal{L}_\eta^{(i)}(\boldsymbol{x})$, with $\mathcal{L}_\eta^{(i)}$ defined in (14). Then*

$$\tilde{\boldsymbol{x}} = \left(\frac{1}{\sum_{i\in\mathcal{F}_j} L(\varphi_t^{(i)})} \sum_{i\in\mathcal{F}_j} L(\varphi_t^{(i)}) x_{k+1,i}^{(j)}\right)_{j\in[1,d]} \tag{18}$$

is optimal in the minimax sense. Furthermore, if any $L(\varphi_t^{(j)})$ is unknown, $\bar{\boldsymbol{x}}_{k+1}$ as defined in (15) is an approximation of the minimax optimum.

Intuitively, Theorem 4.3 proves that an upper bound for $|\sum_{i=1}^{n}\varphi_t^{(i)}(\boldsymbol{x}^*) - \sum_{i=1}^{n}\varphi_t^{(i)}(\tilde{\boldsymbol{x}})|$ exists (with $\boldsymbol{x}^* = \arg\max_{\boldsymbol{x}\in\mathcal{D}}\varphi_t(\boldsymbol{x})$), and that (18) minimizes this upper bound. In addition, if not enough information or computing capacity is available to compute $L(\varphi_t^{(j)})$, using (15) provides a good approximation of the minimax optimal.

5 Asymptotic Optimality

In this section, we provide a regret bound for DuMBO and we establish its asymptotic optimality under the assumptions given above. We start by providing an upper bound on its immediate regret $r_t = f(\boldsymbol{x}^*) - f(\boldsymbol{x}_t)$ for a finite, discrete domain $\mathcal{D}$. Its proof can be found in Appendix F.

Theorem 5.1. *Let $r_t = f(\boldsymbol{x}^*) - f(\boldsymbol{x}^t)$ denote the immediate regret of DuMBO. Let $\delta \in (0,1)$ and $\beta_t = 2\log\left(\frac{|\mathcal{D}|\pi^2 t^2}{6\delta}\right)$. Then $\forall \boldsymbol{x} \in \mathcal{D}, \forall t \in \mathbb{N}$ we have*

$$r_t \leq 2\beta_t^{\frac{1}{2}}\left(a\sum_{i=1}^{n}\left(\sigma_t^{(i)}(\boldsymbol{x}^t)\right)^2 + \frac{1}{4a}\right) \tag{19}$$

with probability at least $1-\delta$.

Table 2: Comparison of four state-of-the-art solutions against two different versions of DuMBO on synthetic and real-world problems. The reported metrics correspond to the minimal regret attained for the synthetic functions, and to the average negative reward for the real-world problems. The best performance metric among all the strategies is written in **bold text**, and the best among the strategies that do not have access to the additive decomposition is underlined. Decomposing BO algorithms can be identified with the prefix "(+)".

Algorithm	Synthetic Functions (d-$\bar{d}$)				Real-World Problems (d-$\bar{d}$)		
Unknown Add. Dec.	SHC (2-2)	Hartmann (6-6)	Powell (24-4)	Rastrigin (100-5)	Cosmo (9-)	WLAN (12-6)	Rover (60-)
SAASBO	0.013	0.89	2,544	1,073	16.55	-116.40	10.82
TuRBO	0.322	1.89	711	1,109	**5.82**	-118.39	**6.06**
(+) ADD-GPUCB	0.102	1.29	10,258	N/A	7.46	-119.05	26.57
(+) DEC-HBO	**0.005**	1.47	9,025	N/A	14.90	-116.58	10.07
(+) DuMBO	0.029	0.76	**542**	1,010	**5.86**	-118.57	6.38
Known Add. Dec.							
(+) ADD-DuMBO	0.102	**0.72**	**542**	**822**	N/A	**-121.06**	N/A

We demonstrate the asymptotic optimality of DuMBO by piggybacking on the asymptotic optimality of DEC-HBO [7]. This is a decomposing BO algorithm with an immediate regret bound of $2\beta_t^{1/2}\sum_{i=1}^{n}\sigma_t^{(i)}(\boldsymbol{x}^t)$ (see Theorem 1 in [7]). Interestingly, Theorem 3.6 directly implies that the immediate regret bound (19) is lower than the immediate regret bound of DEC-HBO. As a consequence, the immediate regret of DuMBO is bounded above by the regret bound of DEC-HBO. This allows us to rely on proofs in [7] to establish some properties of DuMBO. In particular, DEC-HBO is provably asymptotically optimal whether the domain $\mathcal{D}$ is discrete or continuous (see Theorems 2 and 3 in [7]). These results directly apply to DuMBO and imply the following corollary.

Corollary 5.2. *Let $\delta \in (0,1)$ and $R_t = \sum_{k=1}^{t} r_k$ denote the cumulative regret of DuMBO. Then, with probability at least $1-\delta$, there exists a monotonically increasing sequence of $\{\beta_t\}_t$ such that $\beta_t \in \mathcal{O}(\log t)$ and $\lim_{t\to+\infty} R_t/t = 0$.*

6 Performance Experiments

In this section, we detail the experiments carried out to evaluate the empirical performance of DuMBO. Our benchmark comprises four synthetic functions and three real-world experiments. We consider two state-of-the-art decomposing BO algorithms: ADD-GPUCB [6] that assumes that $\bar{d} = 1$, and DEC-HBO [7] that assumes that $\bar{d} \leq 3$. We also consider two state-of-the-art BO algorithms that do not assume an additive decomposition of the objective function: TuRBO [13] and SAASBO [16]. We compare these solutions with two versions of the proposed algorithm: DuMBO that must systematically infer the additive decomposition of f (see Appendix A) and ADD-DuMBO that, conversely, can observe the decomposition if it exists (see Appendix B). Note that the performance evaluation for the early-stopping version of DuMBO (see Section 4.2) is given in Appendix I.

Since BO is often used in the optimization of expensive black-box functions, we are interested in the ability of each algorithm at obtaining good performance in a small number of iterations. Therefore, in every experiment, all the BO algorithms are given 110 iterations to optimize f. Each experiment is repeated 5 independent times. Table 2 gathers the averaged results that were obtained.

6.1 Optimizing Synthetic Functions

In this section, we compare the six BO algorithms mentioned above using four synthetic functions: the 2d Six-Hump Camel (SHC), the 6d Hartmann, the 24d Powell and the 100d Rastrigin. A detailed description of the synthetic functions, as well as the complete set of figures depicting the performance of the BO algorithms can be found in Appendix G.

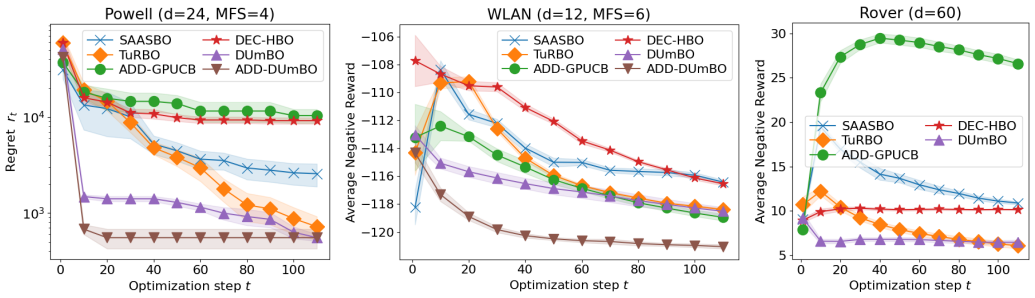


Figure 1: Performance achieved by the BO algorithms listed in Section 6 for (a) the 24d Powell synthetic function, (b) the optimization of the Shannon capacity in a WLAN and (c) the trajectory planning of a rover. The shaded areas indicate the standard error intervals.

Figure 1(a) reports the minimal regrets of the solutions on the Powell function, where $d = 24$ and the MFS $\bar{d} = 4$. Observe that the two decomposing algorithms, ADD-GPUCB and DEC-HBO, obtain the worst minimal regrets. This is because they infer an additive decomposition of f based on the assumption that $\bar{d} \leq 3$ when actually $\bar{d} = 4$. Conversely, DuMBO, which does not make any restrictive assumption on $\bar{d}$, manages to quickly achieve a low regret by inferring an efficient additive decomposition of f. Observe that DuMBO also outperforms SAASBO and TuRBO. Finally, Figure 1(a) shows that, when given access to the true additive decomposition of f, ADD-DuMBO achieves its lowest regret in a lower number of iterations. Note that, among all the BO algorithms tested in the experiments, the two versions of DuMBO are the only ones able to properly infer and/or exploit the additive decomposition of f given its large MFS.

6.2 Solving Real-World Problems

We consider three real-world problems: (a) fine-tuning some cosmological constants to maximize the likelihood of observed astronomical data, (b) controlling the power of devices in a Wireless Local Area Network (WLAN) to maximize its Shannon capacity [36] and (c) the trajectory planning of a rover. The problems, along with a complete set of figures depicting the performance of the tested BO algorithms, are discussed in details in Appendix H.

Figures 1(b) and 1(c) depict the performance of the BO algorithms on problems (b) and (c), where $d = 12$ and 60 respectively. Figure 1(b) shows that DuMBO obtains competitive performance against other state-of-the-art BO algorithms. On this problem as well, the performance of ADD-DuMBO demonstrate that having access, and being able to handle additive decompositions with large MFS, is a significant advantage. As a matter of fact, it allows to outperform other BO algorithms unable to exploit this additional information. Figure 1(c) exhibits patterns similar to Figure 1(a): ADD-GPUCB and DEC-HBO fail to infer an adequate additive decomposition because of the restrictive MFS assumption. Conversely, DuMBO, which does not make such an assumption on the size of the MFS, demonstrates its competitiveness by achieving the best performance along with TuRBO. Note that ADD-DuMBO is not evaluated on problem (c) since its objective function cannot be decomposed.

7 Conclusion

We investigated the benefits of relaxing the restrictive assumptions of low-MFS additive decomposition that limit the applicability domain of state-of-the-art decomposing BO algorithms. As illustrated by Table 1, we chose to optimize the acquisition function with algorithms that scale well with the number of dimensions. This enables us to infer a complex additive decomposition of the objective function f, or to directly exploit it when it is available. To illustrate the effectiveness of such design choices, we proposed DuMBO, an asymptotically optimal decentralized BO algorithm that optimizes f using a tighter decentralized approximation of GP-UCB that requires less exploration than the previously proposed approximations. As demonstrated by Sections 5 and 6, DuMBO is a competitive alternative to state-of-the-art BO algorithms, able to optimize complex objective functions in a

small number of iterations. Compared to other decomposing algorithms, such as ADD-GPUCB and DEC-HBO, DuMBO constitutes a significant improvement, particularly when the decomposition of f has a large MFS with numerous factors or does not exist.

This brings evidence that a more complex model with weaker guarantees on the maximization of the acquisition function φ_t can often represent a better option than a simpler model with stronger guarantees on the maximization of φ_t. For future work, we plan to extend DuMBO to batch mode [37, 38] and to apply it to suitable technological contexts such as networks [24], UAVs [26] or within a robots team [39].

Acknowledgements

This work was supported by the EDIC doctoral program of EPFL and the LABEX MILYON (ANR-10-LABX-0070) of Université de Lyon, within the program "Investissements d'Avenir" (ANR-11-IDEX-0007) operated by the French National Research Agency (ANR).

References

[1] James Bergstra, Daniel Yamins, and David Cox. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In *International conference on machine learning*, pages 115–123. PMLR, 2013.

[2] Daniel Lizotte, Tao Wang, Michael Bowling, and Dale Schuurmans. Automatic gait optimization with gaussian process regression. In *Proceedings of the 20th International Joint Conference on Artifical Intelligence*, IJCAI'07, page 944–949, San Francisco, CA, USA, 2007. Morgan Kaufmann Publishers Inc.

[3] Gregory Hornby, Al Globus, Derek Linden, and Jason Lohn. Automated antenna design with evolutionary algorithms. In *American Institute of Aeronautics and Astronautics*, 2006.

[4] Javier González, Joseph Longworth, David C. James, and Neil D. Lawrence. Bayesian optimization for synthetic gene design. In *NIPS Workshop on Bayesian Optimization in Academia and Industry*, 2014.

[5] Ziyu Wang, Frank Hutter, Masrour Zoghi, David Matheson, and Nando de Freitas. Bayesian optimization in high-dimensions via random embeddings. In *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.

[6] Kirthevasan Kandasamy, Jeff Schneider, and Barnabas Poczos. High dimensional bayesian optimisation and bandits via additive models. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 295–304, Lille, France, 07–09 Jul 2015. PMLR.

[7] Trong Nghia Hoang, Quang Minh Hoang, Ruofei Ouyang, and Kian Hsiang Low. Decentralized high-dimensional bayesian optimization with factor graphs. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018.

[8] Mojmir Mutny and Andreas Krause. Efficient high dimensional bayesian optimization with additivity and quadrature fourier features. In *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018.

[9] Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias W. Seeger. Information-theoretic regret bounds for gaussian process optimization in the bandit setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265, 2012.

[10] Donald R. Jones, Matthias Schonlau, and William J. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.

[11] Jonas Mockus. Application of bayesian approach to numerical methods of global and stochastic optimization. *Journal of Global Optimization*, 4:347–365, 1994.

[12] Peter Auer. Using confidence bounds for exploitation-exploration trade-offs. *J. Mach. Learn. Res.*, 3:397–422, mar 2003.

[13] David Eriksson, Michael Pearce, Jacob Gardner, Ryan D Turner, and Matthias Poloczek. Scalable global optimization via local bayesian optimization. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

[14] Mickaël Binois, David Ginsbourger, and Olivier Roustant. A warped kernel improving robustness in bayesian optimization via random embeddings. In *International Conference on Learning and Intelligent Optimization*, pages 281–286. Springer, 2015.

[15] Ben Letham, Roberto Calandra, Akshara Rai, and Eytan Bakshy. Re-examining linear embeddings for high-dimensional bayesian optimization. *Advances in neural information processing systems*, 33:1546–1558, 2020.

[16] David Eriksson and Martin Jankowiak. High-dimensional Bayesian optimization with sparse axis-aligned subspaces. In *Proceedings of the Thirty-Seventh Conference on Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pages 493–503. PMLR, 27–30 Jul 2021.

[17] Rafael Gómez-Bombarelli, Jennifer N Wei, David Duvenaud, José Miguel Hernández-Lobato, Benjamín Sánchez-Lengeling, Dennis Sheberla, Jorge Aguilera-Iparraguirre, Timothy D Hirzel, Ryan P Adams, and Alán Aspuru-Guzik. Automatic chemical design using a data-driven continuous representation of molecules. *ACS central science*, 4(2):268–276, 2018.

[18] Riccardo Moriconi, Marc Peter Deisenroth, and KS Sesh Kumar. High-dimensional bayesian optimization using low-dimensional feature spaces. *Machine Learning*, 109(9):1925–1943, 2020.

[19] Cheng Li, Sunil Gupta, Santu Rana, Vu Nguyen, Svetha Venkatesh, and Alistair Shilton. High dimensional bayesian optimization using dropout. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 2096–2102, 2017.

[20] Lei Song, Ke Xue, Xiaobin Huang, and Chao Qian. Monte carlo tree search based variable selection for high dimensional bayesian optimization. In *Advances in Neural Information Processing Systems*, 2022.

[21] Zi Wang and Stefanie Jegelka. Max-value entropy search for efficient Bayesian optimization. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3627–3635. PMLR, 06–11 Aug 2017.

[22] Alex Rogers, Alessandro Farinelli, Ruben Stranders, and Nicholas R Jennings. Bounded approximate decentralised coordination via the max-sum algorithm. *Artificial Intelligence*, 175(2):730–759, 2011.

[23] Jacob Gardner, Chuan Guo, Kilian Weinberger, Roman Garnett, and Roger Grosse. Discovering and exploiting additive structure for bayesian optimization. In *Artificial Intelligence and Statistics*, pages 1311–1319. PMLR, 2017.

[24] Anthony Bardou and Thomas Begin. Inspire: Distributed bayesian optimization for improving spatial reuse in dense wlans. In *Proceedings of the 25th International ACM Conference on Modeling Analysis and Simulation of Wireless and Mobile Systems*, pages 133–142, 2022.

[25] Mathieu Bourdeau, Xiao qiang Zhai, Elyes Nefzaoui, Xiaofeng Guo, and Patrice Chatellier. Modeling and forecasting building energy consumption: A review of data-driven techniques. *Sustainable Cities and Society*, 48:101533, 2019.

[26] Lifeng Xie, Jie Xu, and Rui Zhang. Throughput maximization for uav-enabled wireless powered communication networks. *IEEE Internet of Things Journal*, 6(2):1690–1703, 2018.

[27] Kai Wang, Bryan Wilder, Sze-chuan Suen, Bistra Dilkina, and Milind Tambe. Improving gp-ucb algorithm by harnessing decomposed feedback. In *Machine Learning and Knowledge Discovery in Databases*, pages 555–569, Cham, 2020. Springer International Publishing.

[28] Eyke Hüllermeier and Willem Waegeman. Aleatoric and epistemic uncertainty in machine learning: An introduction to concepts and methods. *Machine Learning*, 110(3):457–506, 2021.

[29] Daniel Gabay and Bertrand Mercier. A dual algorithm for the solution of nonlinear variational problems via finite element approximation. *Computers & Mathematics with Applications*, 2(1):17–40, 1976.

[30] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, and Jonathan Eckstein. *Distributed Optimization and Statistical Learning via the Alternating Direction Method of Multipliers*. Now Foundations and Trends, 2011.

[31] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

[32] Athanasios P Liavas and Nicholas D Sidiropoulos. Parallel algorithms for constrained tensor factorization via alternating direction method of multipliers. *IEEE Transactions on Signal Processing*, 63(20):5450–5463, 2015.

[33] Rongjie Lai and Stanley Osher. A splitting method for orthogonality constrained problems. *Journal of Scientific Computing*, 58(2):431–449, 2014.

[34] Rick Chartrand and Brendt Wohlberg. A nonconvex admm algorithm for group sparsity with sparse groups. In *2013 IEEE international conference on acoustics, speech and signal processing*, pages 6009–6013. IEEE, 2013.

[35] Yu Wang, Wotao Yin, and Jinshan Zeng. Global convergence of admm in nonconvex nonsmooth optimization. *J. Sci. Comput.*, 78(1):29–63, jan 2019.

[36] JHB Kemperman. On the shannon capacity of an arbitrary channel. In *Indagationes Mathematicae (Proceedings)*, volume 77, pages 101–115. North-Holland, 1974.

[37] Cheng Li, Paul Resnick, and Qiaozhu Mei. Multiple queries as bandit arms. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 1089–1098, 2016.

[38] Erik A. Daxberger and Bryan Kian Hsiang Low. Distributed batch Gaussian process optimization. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 951–960. PMLR, 06–11 Aug 2017.

[39] Jie Chen, Kian Hsiang Low, and Colin Keng-Yan Tan. Gaussian process-based decentralized data fusion and active sensing for mobility-on-demand system. *arXiv preprint arXiv:1306.1491*, 2013.

[40] Zi Wang, Chengtao Li, Stefanie Jegelka, and Pushmeet Kohli. Batched high-dimensional Bayesian optimization via structural kernel learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3656–3664. PMLR, 06–11 Aug 2017.

[41] David K Duvenaud, Hannes Nickisch, and Carl Rasmussen. Additive gaussian processes. *Advances in neural information processing systems*, 24, 2011.

[42] Christian Robert, George Casella, Christian P Robert, and George Casella. Metropolis–hastings algorithms. *Introducing Monte Carlo Methods with R*, pages 167–197, 2010.

[43] Yuege Xie, Xiaoxia Wu, and Rachel Ward. Linear convergence of adaptive stochastic gradient descent. In *International conference on artificial intelligence and statistics*, pages 1475–1485. PMLR, 2020.

[44] C.-H. Chuang, F. Prada, A. J. Cuesta, D. J. Eisenstein, E. Kazin, N. Padmanabhan, A. G. Sanchez, X. Xu, F. Beutler, M. Manera, D. J. Schlegel, D. P. Schneider, D. H. Weinberg, J. Brinkmann, J. R. Brownstein, and D. Thomas. The clustering of galaxies in the SDSS-III baryon oscillation spectroscopic survey: single-probe measurements and the strong power of f(z) 8(z) on constraining dark energy. *Monthly Notices of the Royal Astronomical Society*, 433(4):3559–3571, jul 2013.

[45] J. Zuntz, M. Paterno, E. Jennings, D. Rudd, A. Manzotti, S. Dodelson, S. Bridle, S. Sehrish, and J. Kowalkowski. CosmoSIS: Modular cosmological parameter estimation. *Astronomy and Computing*, 12:45–59, sep 2015.

[46] The ns3 Project. The Network Simulator ns-3. `https://www.nsnam.org/`. Accessed: 2021-09-30.

[47] Zi Wang, Clement Gehring, Pushmeet Kohli, and Stefanie Jegelka. Batched large-scale bayesian optimization in high-dimensional spaces. In *International Conference on Artificial Intelligence and Statistics*, pages 745–754. PMLR, 2018.

A Inference of the Additive Decomposition

Our decentralized algorithm requires an additive decomposition of the objective function f, as specified in Assumption 3.1 and exploited in Proposition 3.3. If the decomposition is known, it can be directly specified to DuMBO. If the decomposition is unknown, it can be inferred from the data [23, 40]. In this appendix, we briefly discuss how we exploit the approach introduced by [23].

As in [7], let us associate each candidate additive decomposition $\mathcal{A}$ with the kernel of an additive GP [41]. Given k candidates $\mathcal{A}_1, \cdots, \mathcal{A}_k$, we reformulate the acquisition function φ_t as a weighted average with respect to the posterior of each candidate given the dataset $\mathcal{S} = \{(\boldsymbol{x}_i, y_i)\}_{i\in[1,t]}$ composed of the selected input queries and their observed noisy outputs, that is

$$\varphi_t(\boldsymbol{x}) = \sum_{i=1}^{k} p(\mathcal{A}_i|\mathcal{S})\varphi_t^{\mathcal{A}_i}(\boldsymbol{x}) \tag{20}$$

$$= \sum_{i=1}^{k} p(\mathcal{A}_i|\mathcal{S}) \sum_{j=1}^{|\mathcal{A}_i|} \varphi_t^{(j)}(\boldsymbol{x}_{\mathcal{V}_j}) \tag{21}$$

$$\approx \frac{1}{k} \sum_{i=1}^{k} \sum_{j=1}^{|\mathcal{A}_i|} \varphi_t^{(j)}(\boldsymbol{x}_{\mathcal{V}_j}), \tag{22}$$

with $\varphi_t^{\mathcal{A}_i}$ our proposed acquisition function given the additive decomposition $\mathcal{A}_i$, $\varphi_t^{(j)}$ given by (8), (21) following from (20) since the additive decomposition $\mathcal{A}_i$ also provides an additive decomposition of our proposed acquisition function, and (22) following from (21) as demonstrated by [23].

As for the candidates $\mathcal{A}_1, \cdots, \mathcal{A}_k$, they are sampled by Monte-Carlo Markov Chain (MCMC) with the Metropolis-Hastings algorithm [42], starting from the fully dependent decomposition $\mathcal{A}_0 = \{\{1, \cdots, d\}\}$ at $t = 0$. When the decomposition is unknown, at each time step t, k promising decompositions are sampled by MCMC starting from the last sampled decomposition at time step $t-1$, and (22) is maximized by our decentralized algorithm to find a promising input $\boldsymbol{x}$ to query.

B Inference Formulas with Decomposed Output

The decentralized algorithm DuMBO requires an additive decomposition of the objective function f, however we do not assume that the corresponding decomposition of the output is observable. Nevertheless, as demonstrated by [27], having access to such a decomposed output improves the regression capabilities of the surrogate GP model, mostly by reducing the variance of its predictions. In this appendix, we derive the counterparts of the posterior mean (3) and the posterior variance (4) when the output decomposition of f is observable.

Observing the output decomposition of f means that f is now a function $\mathbb{R}^d \to \mathbb{R}^n$, with n the number of factors in its additive decomposition. At a time t, a BO algorithm has no longer access to a t-dimensional output vector $\boldsymbol{y}$, but to a $t \times n$ matrix $\boldsymbol{Y}$. Note that $\boldsymbol{y}$ and $\boldsymbol{Y}$ are linked through the relation $\boldsymbol{Y}\mathbf{1} = \boldsymbol{y}$, with $\mathbf{1}$ the n-dimensional all-1 vector.

Having access to the matrix $\boldsymbol{Y}$ allows to train n different GPs instead of a single one with an additive kernel [41], so that the ith $\mathcal{GP}\left(0, k^{(i)}\left(\boldsymbol{x}_{\mathcal{V}_i}, \boldsymbol{x}'_{\mathcal{V}_i}\right)\right)$ serves as a surrogate model only for the ith factor of the decomposition of f. To condition the ith GP, we consider the data set $\mathcal{S}_i = \left\{\left(\boldsymbol{x}^j_{\mathcal{V}_i}, Y_{j,i}\right)\right\}_{j\in[1,t]}$. Given $\mathcal{S}_i$, the expressions of the posterior mean $\mu^{(i)}_{t+1}$ and the posterior variance $\left(\sigma^{(i)}_{t+1}\right)^2$ are simple instances of the conditioned Gaussian distribution formulas, where

$$\mu^{(i)}_{t+1}(\boldsymbol{x}) = \boldsymbol{k}^{(i)\top}_{\boldsymbol{x}_{\mathcal{V}_i}} \boldsymbol{K}^{-1}_{(i)} \boldsymbol{Y}_{:i}, \tag{23}$$

$$\left(\sigma^{(i)}_{t+1}(\boldsymbol{x})\right)^2 = k^{(i)}\left(\boldsymbol{x}_{\mathcal{V}_i}, \boldsymbol{x}_{\mathcal{V}_i}\right) - \boldsymbol{k}^{(i)\top}_{\boldsymbol{x}_{\mathcal{V}_i}} \boldsymbol{K}^{-1}_{(i)} \boldsymbol{k}^{(i)}_{\boldsymbol{x}_{\mathcal{V}_i}}, \tag{24}$$

with $\boldsymbol{Y}_{:i}$ the ith column of $\boldsymbol{Y}$, $t \times 1$ vectors $\boldsymbol{k}^{(i)}_{\boldsymbol{x}_{\mathcal{V}_i}} = \left(k^{(i)}(\boldsymbol{x}_{\mathcal{V}_i}, \boldsymbol{x}^j_{\mathcal{V}_i})\right)_{j\in[1,t]}$ and $t \times t$ matrices $\boldsymbol{K}_{(i)} = \left(k^{(i)}(\boldsymbol{x}^j_{\mathcal{V}_i}, \boldsymbol{x}^k_{\mathcal{V}_i})\right)_{j,k\in[1,t]}$.

Note that (23) and (24) mainly differ from (3) and (4) by their ability to exploit the inverse of the Gram matrix built only from the ith covariance function $k^{(i)}$, and of course, the outputs of the ith factor of the decomposition $\boldsymbol{Y}_{:i}$.

C Proposed Acquisition Function

In this appendix, we provide the proofs for the approximation of the exploration term in the GP-UCB acquisition function as well as the acquisition function φ_t proposed in the paper. We start by proving the following lemma.

Lemma C.1. *The best linear overestimation of $\sqrt{x}$ (in the least-squares sense) is given by (5), where a is one of the positive roots of the quartic polynomial (6).*

Proof. We want a linear approximation $ax + b$ that consistently overestimates $\sqrt{x}$ over the interval $[v_-, v_+]$. Since $\sqrt{x}$ is concave, the overestimation is ensured if $ax + b - \sqrt{x} = 0$ has at most a single solution in $\mathbb{R}^+$. This can be achieved by adjusting the b parameter so that the polynomial $Q(x) = ax^2 - x + b$ has a single root. The discriminant of Q is $1 - 4ab$, so $\forall a > 0, b = \frac{1}{4a}$ ensures the overestimation of $\sqrt{x}$.

The linear approximation $ax + \frac{1}{4a}$ must also be optimal in the least squares sense. Therefore, we must find

$$\begin{aligned} a^* &= \arg\min_{a \in \mathbb{R}^+} \int_{v_-}^{v_+} \left(\sqrt{u} - \left(au + \frac{1}{4a} \right) \right)^2 du \\ &= \arg\min_{a \in \mathbb{R}^+} \frac{\left[u^3\right]_{v_-}^{v_+}}{3} a^2 - \frac{4\left[u^{\frac{5}{2}}\right]_{v_-}^{v_+}}{5} a + \frac{3\left[u^2\right]_{v_-}^{v_+}}{4} - \frac{\left[u^{\frac{3}{2}}\right]_{v_-}^{v_+}}{3a} + \frac{[u]_{v_-}^{v_+}}{16a^2}. \end{aligned} \tag{25}$$

Differentiating (25) with respect to a and multiplying by a^3 to turn the expression into a polynomial, we get the desired quartic

$$P(a) = \frac{2\left[u^3\right]_{v_-}^{v_+}}{3} a^4 - \frac{4\left[u^{\frac{5}{2}}\right]_{v_-}^{v_+}}{5} a^3 + \frac{\left[u^{\frac{3}{2}}\right]_{v_-}^{v_+}}{3} a - \frac{[u]_{v_-}^{v_+}}{8}.$$

Therefore, a must be one of the positive roots of P. Since $ax + 1/4a$ consistently overestimates $\sqrt{x}$, P has at least one positive root by construction. □

With Lemma C.1, we can prove Proposition 3.5.

Proof. We need to prove that the quartic (6) has a single positive root, which is also the solution of (25). The derivative P' of P has, by construction, the same sign as the second derivative of the expression in (25) and reads

$$P'(a) = \frac{8\left[u^3\right]_{v_-}^{v_+}}{3} a^3 - \frac{12\left[u^{\frac{5}{2}}\right]_{v_-}^{v_+}}{5} a^2 + \frac{\left[u^{\frac{3}{2}}\right]_{v_-}^{v_+}}{3}. \tag{26}$$

The discriminant of (26) can be shown to be always non-positive, yielding that P' has a single real root. Furthermore, it is easy to show that (i) $\lim_{a \to -\infty} P'(a) = -\infty$ and (ii) $\lim_{a \to 0} P'(a) > 0$. Therefore the intermediate value theorem yields that the single real root of P' belongs to $\mathbb{R}^-$. As a consequence, $\forall a \in \mathbb{R}^+, P'(a) > 0$. Therefore, we can conclude that the positive root of P is a minimum.

Furthermore, since $\forall a \in \mathbb{R}^+, P'(a) > 0$, P is increasing and hence cannot have more than one root in $\mathbb{R}^+$. Note that $P(0) < 0$ and $\lim_{a \to +\infty} P(a) = +\infty$, hence P has a unique positive root in $\mathbb{R}^+$ by the intermediate value theorem. As shown by Lemma C.1, this root is the minimizer in (25) and the optimal value for a in the approximation (5). □

We can now prove Theorem 3.6.

Proof. Let

$$S = \left\{ \boldsymbol{x} : \boldsymbol{x} \in \left[\sqrt{v_-^{(1)}}, \sqrt{v_+^{(1)}} \right] \times \cdots \times \left[\sqrt{v_-^{(n)}}, \sqrt{v_+^{(n)}} \right], v_- \leq ||\boldsymbol{x}||_2^2 \leq v_+ \right\}.$$

We need to prove that $\forall \boldsymbol{x} \in S, a||\boldsymbol{x}||_2^2 + 1/4a \leq ||\boldsymbol{x}||_1$. This is equivalent to finding a so that $-a^2||\boldsymbol{x}||_2^2 + a||\boldsymbol{x}||_1 - 1/4 \geq 0$. As a matter of fact, if it exists a fixed a that satisfies the inequality $\forall \boldsymbol{x} \in S$, it is the one computed by the Proposition 3.5, since the approximation is the optimal linear overestimation of $||\boldsymbol{x}||_2$.

We know that $-a^2||\boldsymbol{x}||_2^2 + a||\boldsymbol{x}||_1 - 1/4$ is positive between its roots, which are

$$a_1(\boldsymbol{x}) = \frac{||\boldsymbol{x}||_1 - \sqrt{||\boldsymbol{x}||_1^2 - ||\boldsymbol{x}||_2^2}}{2||\boldsymbol{x}||_2^2}$$
$$a_2(\boldsymbol{x}) = \frac{||\boldsymbol{x}||_1 + \sqrt{||\boldsymbol{x}||_1^2 - ||\boldsymbol{x}||_2^2}}{2||\boldsymbol{x}||_2^2}.$$

In order to ensure the existence of a satisfying the equation for all the elements of S, we need to make sure that

$$\max_{\boldsymbol{x} \in S} a_1(\boldsymbol{x}) \leq \min_{\boldsymbol{x} \in S} a_2(\boldsymbol{x}). \tag{27}$$

To ease the maximization of $a_1(\boldsymbol{x})$, let us consider $\max_{\boldsymbol{x} \in S} \max_{y \in [v_-, ||\boldsymbol{x}||_1^2 - \delta_-^2]} \frac{||\boldsymbol{x}||_1 - \sqrt{||\boldsymbol{x}||_1^2 - y}}{2y}$, with $\delta_-^2 = ||\boldsymbol{x}||_1^2 - ||\boldsymbol{x}||_2^2 = \sum_{i=1}^n \sum_{\substack{j=1 \\ j \neq i}}^n x_i x_j$. A trivial study of the variations of the expression shows that

$$\begin{aligned} \max_{y \in [v_-, ||\boldsymbol{x}||_1^2 - \delta_-^2]} \frac{||\boldsymbol{x}||_1 - \sqrt{||\boldsymbol{x}||_1^2 - y}}{2y} &= \frac{||\boldsymbol{x}||_1 - \delta_-}{2(||\boldsymbol{x}||_1^2 - \delta_-^2)} \\ &= \frac{1}{2(||\boldsymbol{x}||_1 + \delta_-)} \end{aligned}$$

Therefore, $\max_{\boldsymbol{x} \in S} \frac{1}{2(||\boldsymbol{x}||_1 + \delta_-)} \leq \frac{1}{2(\sqrt{v_-} + \delta_-)}$ and $\delta_-^2 = \sum_{i=1}^n \sum_{\substack{j=1 \\ j \neq i}}^n \sqrt{v_-^{(i)} v_-^{(j)}}$.

Similarly, we study the variation of $\frac{||\boldsymbol{x}||_1 + \sqrt{||\boldsymbol{x}||_1^2 - y}}{2y}$ for $y \in [v_-, ||\boldsymbol{x}||_1^2 - \delta_+^2]$. It is trivial to show that

$$\begin{aligned} \min_{y \in [v_-, ||\boldsymbol{x}||_1^2 - \delta_+^2]} \frac{||\boldsymbol{x}||_1 + \sqrt{||\boldsymbol{x}||_1^2 - y}}{2y} &= \frac{||\boldsymbol{x}||_1 + \delta_+}{2(||\boldsymbol{x}||_1^2 - \delta_+^2)} \\ &= \frac{1}{2(||\boldsymbol{x}||_1 - \delta_+)} \end{aligned}$$

Therefore, $\min_{\boldsymbol{x} \in S} \frac{1}{2(||\boldsymbol{x}||_1 - \delta_+)} \geq \frac{1}{2(\sqrt{v_+} - \delta_+)}$, with $\delta_+ \geq \delta_-$. We can now rewrite our criterion (27) as $\sqrt{v_-} + \delta_- \geq \sqrt{v_+} - \delta_+$, and we replace δ_+ by δ_- for the sake of simplicity. This leads to the desired criterion expressed only with the variance bounds

$$\sqrt{v_+} \leq \sqrt{v_-} + 2\delta_- \tag{28}$$

Therefore, whenever (28) holds, $a||\boldsymbol{x}||_2^2 + \frac{1}{4a} \leq ||\boldsymbol{x}||_1$, which is the desired result. □

D Time Complexity of DuMBO

In this short section, we provide a time complexity analysis for DuMBO. The analysis assumes that a gradient ascent performs $\mathcal{O}\left(\zeta^{-1}\right)$ steps for a desired accuracy ζ [43] and ADMM converges in at most N_A steps. We also denote by $d^{(i)}$ the factor size of the ith factor in the decomposition, used by the local acquisition function $\varphi_t^{(i)}$. Note that, within the factor graph of φ_t, n factor nodes and d variable nodes work concurrently to run ADMM in a decentralized fashion. We provide the time complexities for the two types of nodes in this appendix (note that the communication costs between factor nodes and variable nodes have been neglected for the clarity of the analysis).

Factor node. For a factor node i, it is known that, at iteration t, the time complexity of the inference with a GP is $\mathcal{O}\left(t^3 d^{(i)}\right)$, where t denotes the number of previous observations. Thus, the time complexity of evaluating (14) is $\mathcal{O}\left(t^3 d^{(i)}\right)$. Since the evaluation is required $\mathcal{O}\left(\zeta^{-1}\right)$ times by the gradient ascent, the time complexity of finding $x_{k+1}^{(i)}$ is $\mathcal{O}\left(\zeta^{-1} t^3 d_m^{(i)}\right)$. A factor node also needs to compute $\boldsymbol{\lambda}_{k+1}^{(i)}$, which is $\mathcal{O}\left(d^{(i)}\right)$. Since the factor node is called at least once and at most N_A times for ADMM to converge, the time complexity of a factor node is $\mathcal{O}\left(d^{(i)} \zeta^{-1} t^3 N_A\right)$.

Variable node. A variable node j is simply in charge of collecting messages from $|\mathcal{F}_j|$ factor nodes, and to aggregate them into $\bar{x}_{k+1,j}$ by averaging. Its time complexity is therefore $\mathcal{O}\left(|\mathcal{F}_j|\right)$.

E Early-stopping guarantee

This appendix contains the proofs of Proposition 4.2 and Theorem 4.3. Let us start by proving Proposition 4.2.

Proof. We want to show that $\varphi_t^{(i)} = \mu_t^{(i)} + a\beta_t^{\frac{1}{2}} {\sigma_t^{(i)}}^2$ is Lipschitz continuous. This is true if $||\nabla \varphi_t^{(i)}||_2$ is bounded. Replacing $\mu_t^{(i)}$ and ${\sigma_t^{(i)}}^2$ by their expressions (3) and (4), and differentiating with respect to $\boldsymbol{x}$ shows that we need to bound

$$\begin{aligned} ||\nabla \varphi_t^{(i)}||_2 &= ||\nabla \boldsymbol{k}_{\boldsymbol{x}}^{(i)\top} \boldsymbol{K}^{-1} \left(\boldsymbol{y} - 2a\beta_t^{\frac{1}{2}} \boldsymbol{k}_{\boldsymbol{x}}^{(i)}\right)||_2 \\ &\leq ||\nabla \boldsymbol{k}_{\boldsymbol{x}}^{(i)\top}||_2 ||\boldsymbol{K}^{-1}||_2 ||\boldsymbol{y} - 2a\beta_t^{\frac{1}{2}} \boldsymbol{k}_{\boldsymbol{x}}^{(i)}||_2 \end{aligned} \tag{29}$$

Let us upper bound properly all the terms in (29). By Assumption 3.4, we know that $\forall j \in [1,t], v_-^{(i)} \leq k^{(i)}(\boldsymbol{x}, \boldsymbol{x}^j) \leq v_+^{(i)}$. Similarly, $y_t^- \leq y_j \leq y_t^+$, with $y_t^- = \min_{j\in[1,t]} y_j$ and $y_t^+ = \max_{j\in[1,t]} y_j$. Therefore, denoting $M_t^{(i)} = \max\left(|y_t^+ - 2a\beta_t^{\frac{1}{2}} v_-^{(i)}|, |y_t^- - 2a\beta_t^{\frac{1}{2}} v_+^{(i)}|\right)$, we have $||\boldsymbol{y} - 2a\beta_t^{\frac{1}{2}} \boldsymbol{k}_{\boldsymbol{x}}^{(i)}||_2 \leq \sqrt{t} M_t^{(i)}$. Moreover, by the Raleigh–Ritz theorem, it is known that the spectral norm of a symmetric positive semi-definite matrix coincides with its spectral radius (*i.e.*, its largest eigenvalue). Therefore, $||\boldsymbol{K}^{-1}||_2 = \rho\left(\boldsymbol{K}^{-1}\right)$. Finally, we upper bound $||\nabla \boldsymbol{k}_{\boldsymbol{x}}^{(i)\top}||_2$ using the definition of the spectral norm

$$\begin{aligned} ||\nabla \boldsymbol{k}_{\boldsymbol{x}}^{(i)\top}||_2 &= \sup_{\boldsymbol{z}\in\mathbb{R}^t} \frac{||\nabla \boldsymbol{k}_{\boldsymbol{x}}^{(i)\top} \boldsymbol{z}||_2}{||\boldsymbol{z}||_2} \\ &\leq \sup_{\boldsymbol{z}\in\mathbb{R}^t} \frac{\sum_{j=1}^t ||\nabla \boldsymbol{k}^{(i)}(\boldsymbol{x}, \boldsymbol{x}^j)||_2 |z_j|}{\frac{1}{\sqrt{t}} \sum_{j=1}^t |z_j|} \\ &\leq \sup_{\boldsymbol{z}\in\mathbb{R}^t} \frac{\sqrt{t} L\left(k^{(i)}\right) \sum_{j=1}^t |z_j|}{\sum_{j=1}^t |z_j|} \\ &= \sqrt{t} L\left(k^{(i)}\right) \end{aligned} \tag{30}$$

where (30) follows from $k^{(i)}$ being Lipschitz continuous with Lipschitz constant $L\left(k^{(i)}\right)$ according to Assumption 4.1.

Combining all these upper bounds, we have an upper bound for the gradient of $\varphi_t^{(i)}$

$$||\nabla \varphi_t^{(i)}||_2 \leq tL\left(k^{(i)}\right) \rho\left(\boldsymbol{K}^{-1}\right) M_t^{(i)} \tag{31}$$

with $M_t^{(i)} = \max\left(|y_{i,t}^+ - 2a\beta_t^{\frac{1}{2}} v_-^{(i)}|, |y_{i,t}^- - 2a\beta_t^{\frac{1}{2}} v_+^{(i)}|\right)$, which is the desired result. □

We now prove Theoreom 4.3, which claims that a solution from an early-stopped version of DuMBO is still optimal in a weaker sense.

Proof. Denoting $\boldsymbol{x}^* = \arg\max_{\boldsymbol{x}\in\mathcal{D}} \sum_{i=1}^n \varphi_t^{(i)}(\boldsymbol{x}_{\mathcal{V}_i})$ and having $\left\{\boldsymbol{x}_{k+1}^{(i)}\right\}_{i\in[1,n]}$, let us try to find a closed form for the upper bound of $G = |\sum_{i=1}^n \varphi_t^{(i)}(\boldsymbol{x}^*_{\mathcal{V}_i}) - \sum_{i=1}^n \varphi_t^{(i)}(\tilde{\boldsymbol{x}}_{\mathcal{V}_i})|$, before building $\tilde{\boldsymbol{x}}$ that minimizes this upper bound. Since Proposition 4.2 holds, each $\varphi_t^{(i)}$ is Lipschitz and we have

$$\begin{aligned} G &\leq \sum_{i=1}^n |\varphi_t^{(i)}(\boldsymbol{x}^*_{\mathcal{V}_i}) - \varphi_t^{(i)}(\tilde{\boldsymbol{x}}_{\mathcal{V}_i})| \\ &\leq \sum_{i=1}^n L(\varphi_t^{(i)})||\boldsymbol{x}^*_{\mathcal{V}_i} - \tilde{\boldsymbol{x}}_{\mathcal{V}_i}||_2 \\ &\leq \sum_{i=1}^n L(\varphi_t^{(i)}) \left(||\boldsymbol{x}^*_{\mathcal{V}_i} - \boldsymbol{x}_{k+1}^{(i)}||_2 + ||\boldsymbol{x}_{k+1}^{(i)} - \tilde{\boldsymbol{x}}_{\mathcal{V}_i}||_2\right). \end{aligned} \tag{32}$$

where the last inequality (32) follows from the triangle inequality. Since $\boldsymbol{x}_{k+1}^{(i)} = \arg\max_{\boldsymbol{x}} \mathcal{L}_\eta^{(i)}$, $\sum_{i=1}^n L(\varphi_t^{(i)})||\boldsymbol{x}^*_{\mathcal{V}_i} - \boldsymbol{x}_{k+1}^{(i)}||_2$ will get increasingly smaller as ADMM iterates, provided that $\varphi_t^{(i)}$ are restricted proxy-regular (as required by [35]). From (32), we see that $\tilde{\boldsymbol{x}}$ must satisfy $\tilde{\boldsymbol{x}} = \arg\min_{\boldsymbol{x}} \sum_{i=1}^n L(\varphi_t^{(i)})||\boldsymbol{x}_{k+1}^{(i)} - \boldsymbol{x}_{\mathcal{V}_i}||_2$ to minimize the upper bound. This is equivalent to finding $\arg\min_{\boldsymbol{x}} \Psi(\boldsymbol{x}) = \sum_{i=1}^n L(\varphi_t^{(i)})||\boldsymbol{x}_{k+1}^{(i)} - \boldsymbol{x}_{\mathcal{V}_i}||_2^2$. Developing this expression, we have

$$\Psi(\tilde{\boldsymbol{x}}) = \sum_{i=1}^{n_\xi} \sum_{j\in\mathcal{V}_i} L(\varphi_t^{(i)}) \left(x_{k+1,j}^{(i)} - \tilde{x}_j\right)^2. \tag{33}$$

Differentiating (33) with respect to $\tilde{x}_j$, we get

$$\frac{\partial\Psi}{\partial\tilde{x}_j} = 2\sum_{i\in\mathcal{F}_j} L(\varphi_t^{(i)}) \left(\tilde{x}_j - x_{k+1,j}^{(i)}\right).$$

Solving $\partial\Psi/\partial\tilde{x}_j = 0$ is straightforward with the Hessian $H(\Psi)(\tilde{\boldsymbol{x}})$ positive definite, and leads to the minimum

$$\tilde{\boldsymbol{x}} = \left(\frac{1}{\sum_{j\in\mathcal{F}_i} L(\varphi_t^{(j)})} \sum_{j\in\mathcal{F}_i} L(\varphi_t^{(j)}) x_{k+1,i}^{(j)}\right)_{i\in[1,d]} \tag{34}$$

which is the desired result. Note that if $L(\varphi_t^{(j)})$ cannot be computed explicitly in (34), we can upper bound $L(\varphi_t^{(j)})$ by $\max_{j\in[1,n]} L(\varphi_t^{(j)})$ which then cancels out in the numerator and denominator of (34) to become

$$\tilde{\boldsymbol{x}} = \left(\frac{1}{|\mathcal{F}_i|} \sum_{j\in\mathcal{F}_i} x_{k+1,i}^{(j)}\right)_{i\in[1,d]}. \tag{35}$$

Therefore, (35) is a decent approximation of the minimax optimal. Note that, as stated in the theorem, this approximation is exactly $\bar{\boldsymbol{x}}_{k+1}$ as defined in (15). □

F Immediate Regret Bound

In this section, we discuss the asymptotic optimality of DuMBO and provide the proof for Theorem 5.1. We start by proving the following inequality linking $f(\boldsymbol{x})$ with the posterior mean and variance of f.

Lemma F.1. *Pick $\delta \in (0,1)$ and let $\beta_t = 2\log\left(\frac{|\mathcal{D}|\pi^2 t^2}{6\delta}\right)$. Then, with probability at least $1-\delta$,*

$$|f(\boldsymbol{x}) - \mu_t(\boldsymbol{x})| \leq \beta_t^{\frac{1}{2}} \left(a\sigma_t^2(\boldsymbol{x}) + \frac{1}{4a}\right) \tag{36}$$

for all $x \in \mathcal{D}$ and $t \in \mathbb{N}$, $\mu_t(\boldsymbol{x})$ and $\sigma_t^2(\boldsymbol{x})$ defined through our proposed decomposition in Proposition 3.3.

Proof. For all $\boldsymbol{x} \in \mathcal{D}$ and $t \in \mathbb{N}$, we have $f(\boldsymbol{x}) \sim \mathcal{N}\left(\mu_t(\boldsymbol{x}), \sigma_t^2(\boldsymbol{x})\right)$. Defining $s_t(\boldsymbol{x}) = \frac{f(\boldsymbol{x})-\mu_t(\boldsymbol{x})}{\sigma_t(\boldsymbol{x})}$, we know that $s_t(\boldsymbol{x}) \sim \mathcal{N}(0,1)$. Therefore we have successively that

$$\begin{aligned}
\Pr\left(|s_t(\boldsymbol{x})| \leq \beta_t^{\frac{1}{2}}\right) &\geq 1 - e^{-\frac{\beta_t}{2}} \\
\Pr\left(|f(\boldsymbol{x}) - \mu_t(\boldsymbol{x})| \leq \beta_t^{\frac{1}{2}} \sigma_t(\boldsymbol{x})\right) &\geq 1 - e^{-\frac{\beta_t}{2}} \\
\Pr\left(|f(\boldsymbol{x}) - \mu_t(\boldsymbol{x})| \leq \beta_t^{\frac{1}{2}} \left(a\sigma_t^2(\boldsymbol{x}) + \frac{1}{4a}\right)\right) &\geq 1 - e^{-\frac{\beta_t}{2}} \qquad (37)
\end{aligned}$$

where the last inequality (37) follows from $\sqrt{x} \leq ax + \frac{1}{4a}$ (see Proposition 3.5). The inequality (37) holds for one single pair $(t, \boldsymbol{x})$. Applying the union bound for all pairs in $\mathbb{N} \times \mathcal{D}$, we have $\forall t \in \mathbb{N}, \forall \boldsymbol{x} \in \mathcal{D}$

$$\Pr\left(|f(\boldsymbol{x}) - \mu_t(\boldsymbol{x})| \leq \beta_t^{\frac{1}{2}} \left(a\sigma_t^2(\boldsymbol{x}) + \frac{1}{4a}\right)\right) \geq 1 - |\mathcal{D}| \sum_{t=1}^{+\infty} e^{-\frac{\beta_t}{2}}. \qquad (38)$$

Pick $\delta \in (0,1)$ and let $\beta_t = 2\log\left(\frac{|\mathcal{D}|\pi^2 t^2}{6\delta}\right)$. Then,

$$\begin{aligned}
|\mathcal{D}| \sum_{t=1}^{+\infty} e^{-\frac{\beta_t}{2}} &= |\mathcal{D}| \sum_{t=1}^{+\infty} e^{-\log\left(\frac{|\mathcal{D}|\pi^2 t^2}{6\delta}\right)} \\
&= \frac{6\delta}{\pi^2} \sum_{t=1}^{+\infty} \frac{1}{t^2} \\
&= \delta.
\end{aligned}$$

Therefore, (38) becomes

$$\Pr\left(|f(\boldsymbol{x}) - \mu_t(\boldsymbol{x})| \leq \beta_t^{\frac{1}{2}} \left(a\sigma_t^2(\boldsymbol{x}) + \frac{1}{4a}\right)\right) \geq 1 - \delta \qquad (39)$$

which is the desired result. □

We are now ready to bound the immediate regret $r_t = f(\boldsymbol{x}^*) - f(\boldsymbol{x}^t)$ and prove Theorem 5.1.

Proof. By definition, $\boldsymbol{x}^t = \arg\max_{\boldsymbol{x}} \sum_{i=1}^n \varphi_t^{(i)}(\boldsymbol{x}_{\mathcal{V}_i})$. Therefore, $\sum_{i=1}^n \varphi_t^{(i)}(\boldsymbol{x}^t_{\mathcal{V}_i}) \geq \sum_{i=1}^n \varphi_t^{(i)}(\boldsymbol{x}^*_{\mathcal{V}_i})$. Developing the left hand side of this inequality with the expression of φ_t and adding $\frac{\beta_t^{\frac{1}{2}}}{4a}$ on both sides, we have

$$\begin{aligned}
\frac{\beta_t^{\frac{1}{2}}}{4a} + \sum_{i=1}^n \mu^{(i)}(\boldsymbol{x}^t_{\mathcal{V}_i}) + a\beta_t^{\frac{1}{2}} \sigma_t^{(i)}(\boldsymbol{x}^t_{\mathcal{V}_i})^2 &\geq \frac{\beta_t^{\frac{1}{2}}}{4a} + \sum_{i=1}^n \varphi_t^{(i)}(\boldsymbol{x}^*_{\mathcal{V}_i'}) \\
&\geq f(\boldsymbol{x}^*) \qquad (40)
\end{aligned}$$

with (40) following from (39) with high probability. We can now upper bound the immediate regret r_t

$$\begin{aligned}
r_t &= f(\boldsymbol{x}^*) - f(\boldsymbol{x}^t) \\
&\leq \frac{\beta_t^{\frac{1}{2}}}{4a} + \sum_{i=1}^n \mu^{(i)}(\boldsymbol{x}^t_{\mathcal{V}_i}) + a\beta_t^{\frac{1}{2}} \sigma_t^{(i)}(\boldsymbol{x}^t_{\mathcal{V}_i})^2 - f(\boldsymbol{x}^t) \\
&= \sum_{i=1}^n \mu^{(i)}(\boldsymbol{x}^t_{\mathcal{V}_i}) - f(\boldsymbol{x}^t) + \beta_t^{\frac{1}{2}} \left(a \sum_{i=1}^n \sigma_t^{(i)}(\boldsymbol{x}^t_{\mathcal{V}_i})^2 + \frac{1}{4a}\right) \\
&= \mu_t(\boldsymbol{x}^t) - f(\boldsymbol{x}^t) + \beta_t^{\frac{1}{2}} \left(a\sigma_t^2(\boldsymbol{x}^t) + \frac{1}{4a}\right) \qquad (41)
\end{aligned}$$

Combining the conclusion of Lemma F.1 with (41), we get

$$\Pr\left(r_t \leq 2\beta_t^{\frac{1}{2}}\left(a\sigma_t^2(\boldsymbol{x}^t) + \frac{1}{4a}\right)\right) \geq 1-\delta, \tag{42}$$

which is the desired result. □

G Synthetic Functions

In this section, we describe the synthetic functions constituting our benchmark in Section 6.1.

G.1 Six-Hump Camel Function

The Six-Hump Camel function is a 2-dimensional function defined by

$$f(x_1, x_2) = \left(-4 + 2.1x_1^2 - \frac{x_1^4}{3}\right)x_1^2 - x_1x_2 + \left(4 - 4x_2^2\right)x_2^2. \tag{43}$$

It is composed of $n = 3$ factors, with a MFS $\bar{d} = 2$. In our experiment, we optimize it on the rectangle $\mathcal{D} = [-3, 3] \times [-2, 2]$. It has 6 local maxima, two of which are global with $f(\boldsymbol{x}^*) = 1.0316$.

G.2 Hartmann Function

The Hartmann function is a 6-dimensional function defined by

$$f(\boldsymbol{x}) = \sum_{i=1}^{4} \alpha_i \exp\left(-\sum_{j=1}^{6} A_{ij}\left(x_j - P_{ij}\right)^2\right), \tag{44}$$

with $\boldsymbol{\alpha} = (\alpha_i)_{i\in[1,4]}$, $\boldsymbol{A} = (A_{ij})_{(i,j)\in[1,4]\times[1,6]}$ and $\boldsymbol{P} = (P_{ij})_{(i,j)\in[1,4]\times[1,6]}$ given as constants.

It is composed of $n = 4$ factors, with a MFS $\bar{d} = 6$. In our experiment, we optimize it on the hypercube $\mathcal{D} = [0, 1]^6$. It has 6 local maxima and a global maximum with $f(\boldsymbol{x}^*) = 10.5364$.

G.3 Powell Function

The Powell function is a function of an arbitrary number $d = 4k$ of dimensions, defined by

$$f(\boldsymbol{x}) = -\sum_{i=1}^{d/4}(x_{4i-3}+10x_{4i-2})^2+5(x_{4i-1}-x_{4i})^2+(x_{4i-2}-2x_{4i-1})^4+10(x_{4i-3}-x_{4i})^4. \tag{45}$$

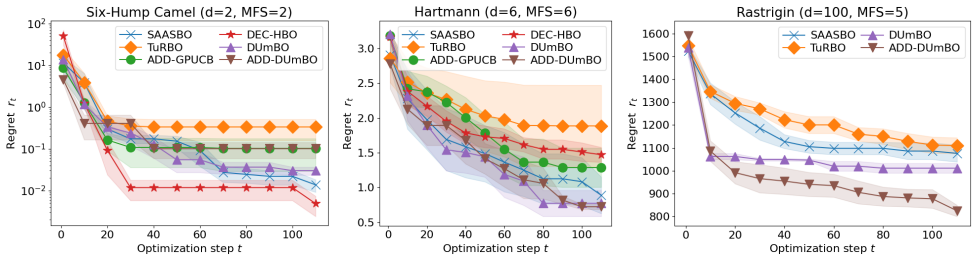
We chose to set $k = 6$, so that the resulting Powell function lives in a $d = 24$ dimensional space. It is composed of $n = 6$ factors, with a MFS $\bar{d} = 4$. In our experiment, we optimize it on the hypercube $\mathcal{D} = [-4, 5]^{24}$. It has a global maximum at $\boldsymbol{x}^* = \boldsymbol{0}$, with $f(\boldsymbol{x}^*) = 0$.

G.4 Rastrigin Function

The Rastrigin function is a function of an arbitrary number d of dimensions, defined by

$$f(\boldsymbol{x}) = -10d - \sum_{i=1}^{d} x_i^2 - 10\cos\left(2\pi x_i\right). \tag{46}$$

We chose to set $d = 100$. We also chose to aggregate some factors to make the optimization problem harder. The resulting Rastrigin function is composed of $n = 20$ factors, with a MFS $\bar{d} = 5$. In our experiment, we optimize it on the hypercube $\mathcal{D} = [-5.12, 5.12]^{100}$. It has multiple, regularly distributed local maxima, with a global maximum at $\boldsymbol{x}^* = \boldsymbol{0}$ and $f(\boldsymbol{x}^*) = 0$.



(a) (b) (c)

Figure 2: Performance achieved by the studied BO algorithms for (a) the 2d Six-Hump Camel function, (b) the 6d Hartmann function and (c) the 100d Rastrigin function. The shaded areas indicate the standard error intervals.

G.5 Additional Figures

Figure 2 depicts the performance of the studied BO algorithms on the synthetic functions not discussed in Section 6.1.

Figure 2(a) reports the minimal regrets achieved by the solutions on the Six-Hump Camel (SHC) function. Observe that in this specific example DEC-HBO obtains the best performance. This is due to the simplicity of SHC. In fact, the SHC function satisfies all the assumptions made by DEC-HBO: a MFS lower than 3 and a sparse factor graph. In this case, the variant of the max-sum algorithm used by DEC-HBO is guaranteed to query $\arg\max \varphi_t$ at each time step t. Since DuMBO does not offer stronger maximization guarantees in that case, it is outperformed by DEC-HBO. Still, note that it exhibits competitive performance when compared to the remaining BO algorithms.

Figure 2(b) and 2(c) depict dynamics similar to Figure 1(a). In both cases, the ability to infer / exploit a complex additive decomposition gives DuMBO a decisive advantage against the other BO algorithms. As a consequence, it manages to outperform them, even in very high dimensional input spaces (see Figure 2(c)). Note that ADD-GPUCB and DEC-HBO were not evaluated on the Rastrigin function, as their execution time exceeded 24 hours because of the large dimensionality of the function.

H Real-World Problems

In this appendix, we describe the real-world problems constituting our benchmark in Section 6.2.

H.1 Cosmological Constants

The cosmological constants problem consists in fine-tuning an astrophysics tool to optimize the likelihood of some observed data. We chose to compute the likelihood of the galaxy clustering [44] from the Data Release 9 (DR9) CMASS sample of the SDSS-III Baryon Oscillation Spectroscopic Survey (BOSS). To compute the likelihood, we instrumented the cosmological parameter estimation code CosmoSIS [45][1].

We used nine cosmological constants in our optimization task, going from the Hubble's constant to the mass of the neutrinos. If a BO algorithm provided a set of inconsistent cosmological constants, a likelihood of $y = -60$ was returned.

Note that similar experiments were described in other works, such as [6, 13]. However, they were conducted on another, older dataset, with a deprecated NASA simulator[2]. This makes the conducted experiments painful to reproduce on a modern computer. Hopefully, CosmoSIS is well documented and easier to install and instrument, so we conducted our experiment with CosmoSIS to make it easier to replicate.

[1] `https://cosmosis.readthedocs.io/en/latest/reference/standard_library/BOSS.html`
[2] `https://lambda.gsfc.nasa.gov/toolbox/lrgdr/`

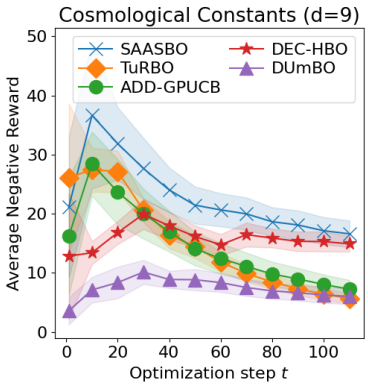


Figure 3: Performance of the studied BO algorithms on the cosmological constants fine-tuning problem.

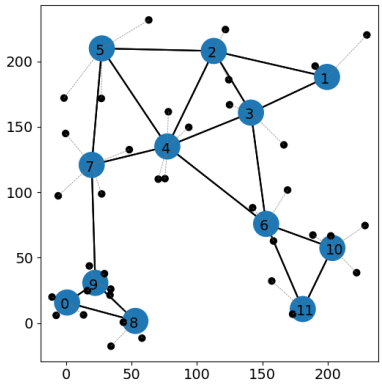


Figure 4: The WLAN topology used in the Shannon capacity optimization experiment. The end-users are depicted as black dots, the nodes as numbered blue circles and the associations between end-users and nodes as thin gray lines. Two nodes are connected with a black line if they are within the radio range of each other.

Figure 3 depicts the performance of the described BO algorithms on this problem. Note that, since the objective function does not have an additive decomposition, ADD-DuMBO cannot be evaluated. Although the objective function does not have an additive decomposition, DuMBO demonstrates its competitiveness by achieving the best performance, along with ADD-GPUCB and TuRBO.

H.2 Shannon Capacity of a WLAN

The Shannon capacity [36] sets a theoretical upper bound on the throughput of a wireless communication, depending on the Signal-to-Interference plus Noise Ratio (SINR) of the communication. Denoting by $S_{i,j}$ the SINR between two wireless devices i and j communicating on a radio channel of bandwidth W (in Hz), the Shannon capacity $C(S_{i,j})$ (in bits) is defined by

$$C(S_{i,j}) = W \log_2 (1 + S_{i,j}) \,. \tag{47}$$

In this problem, we study a Wireless Local Area Network (WLAN) with end-users associated to nodes streaming a continuous, large amount of data. The WLAN topology is depicted in Figure 4. It is populated with 36 end-users, each one associated to one of the 12 depicted nodes. Note that each node is within the radio range of at least two other nodes. This creates interference and, consequently, reduces the SINRs between nodes and end-users.

Each node has an adjustable transmission power $x_i \in [10^{0.1}, 10^{2.5}]$ in mW (milliwatts). This task consists in jointly optimizing the Shannon capacity (47) of each pair (node, associated end-user) by tuning the transmission power of the nodes. That is, the objective function f is a 12-dimensional

function defined by

$$f(\boldsymbol{x}) = \sum_{i=1}^{12} \sum_{j \in \mathcal{N}_i} C(S_{i,j}), \tag{48}$$

with $\mathcal{N}_i$ the set of end-users associated to node i.

A difficult trade-off needs to be found because a node cannot simply use the maximum transmission power as this would cause a lot of interference for the neighboring nodes. Given a configuration $\boldsymbol{x} \in \mathcal{D} = [10^{0.1}, 10^{2.5}]^{12}$, the SINRs are provided by the well-recognized network simulator ns-3 [46] that reliably reproduces the WLAN internal dynamics. The additive decomposition comprises $n = 12$ factors, with a MFS of $\bar{d} = 5$, obtained by making the reasonable assumption that only the neighboring nodes of node i (*i.e.* those within the radio range of node i) are creating interference for the communications of node i.

H.3 Rover Trajectory Planning

This problem was also considered by [13, 47]. The goal is to optimize the trajectory of a rover from a starting point $\boldsymbol{s} \in [0, 1]^2$ to a target $\boldsymbol{t} \in [0, 1]^2$, over a rough terrain.

The trajectory is defined by a vector of $d = 60$ dimensions, reshaped into 30 2-d points in $[0, 1]$. A B-spline is fitted to these 30 points, determining the trajectory of the rover. The objective function to optimize is

$$f(\boldsymbol{x}) = -c(\boldsymbol{x}) - 10(||\boldsymbol{x}_{0,1} - \boldsymbol{s}||_1 + ||\boldsymbol{x}_{59,60} - \boldsymbol{t}||_1), \tag{49}$$

with $c(\boldsymbol{x})$ the cost of the trajectory, obtained by integrating the terrain roughness function over the B-spline, and the two L_1-norms serving as incentives to start the trajectory near $\boldsymbol{s}$, and to end it near $\boldsymbol{t}$.

I Performance Evaluation of the Early-Stopped Version of DuMBO

In this section, we evaluate the performance of ES-DuMBO and ES-ADD-DuMBO, the early-stopped versions of DuMBO and ADD-DuMBO, respectively. Recall that the early-stopping procedure and guarantees are described in Section 4.2. In this appendix, the solutions are early-stopped at the end of the very first ADMM iteration. For the sake of readability, we only depict the performance of DuMBO and ADD-DuMBO to compare them with their early-stopped versions.

Figure 5 depicts the performance of the early-stopped versions on the synthetic functions described in Appendix G. Except for the SHC function (Figure 5(a)), the same dynamic can be observed. The early-stopped versions ES-DuMBO and ES-ADD-DuMBO obtain slightly worse performance than their counterparts DuMBO and ADD-DuMBO. However, they remain very competitive, as they outperform the state-of-the-art decomposing BO algorithms in 3 out of 4 synthetic experiments. Regarding the SHC function, the early-stopped versions achieve better performance than their counterparts. ES-DuMBO even achieves similar performance than DEC-HBO. As an extension to this article, we plan to work more on explaining this observation conceptually.

Figure 6 depicts the performance of the early-stopped versions on the real-world problems considered in Appendix H. The three experiments report the same results as those observed for the synthetic functions in Figure 5. ES-DuMBO and ES-ADD-DuMBO perform slightly worse than their counterparts, but they remain very competitive as they outperform DEC-HBO in all three experiments, and ADD-GPUCB on the rover trajectory planning problem (Figure 6(c)). In the two remaining problems (Figures 6(a), 6(b)), the early-stopped versions achieve better than or equivalent performance as ADD-GPUCB.

These result demonstrate that, although the early-stopped version of DuMBO provides only minimax guarantees on the maximization of φ_t, its excellent empirical performance along with its lower execution time (compared to DuMBO) make it a very interesting solution in technological contexts that cannot afford high computing capabilities.

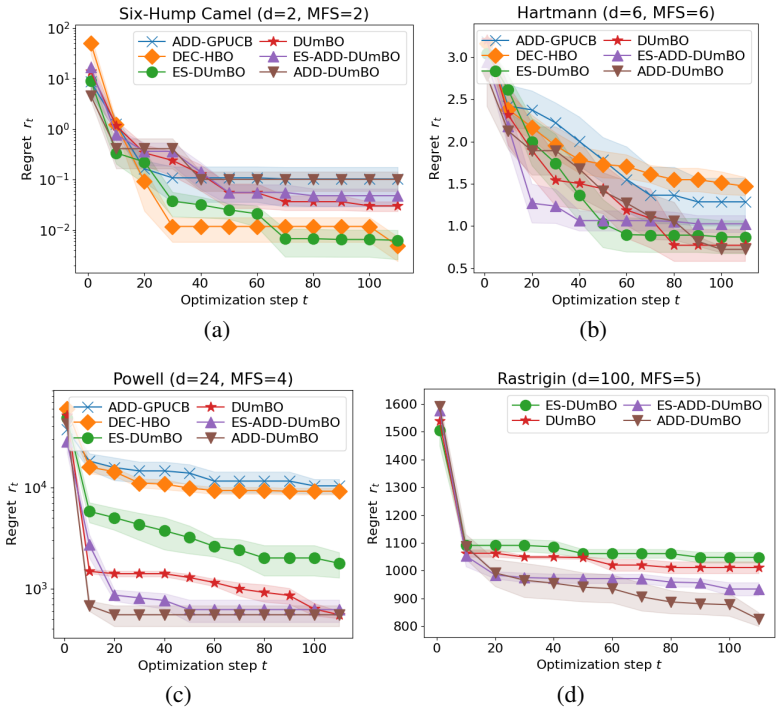


Figure 5: Performance achieved by the decomposing BO algorithms and the early-stopped versions of DuMBO for (a) the 2d Six-Hump Camel function, (b) the 6d Hartmann function, (c) the 24d Powell function and (d) the 100d Rastrigin function. The shaded areas indicate the standard error intervals.

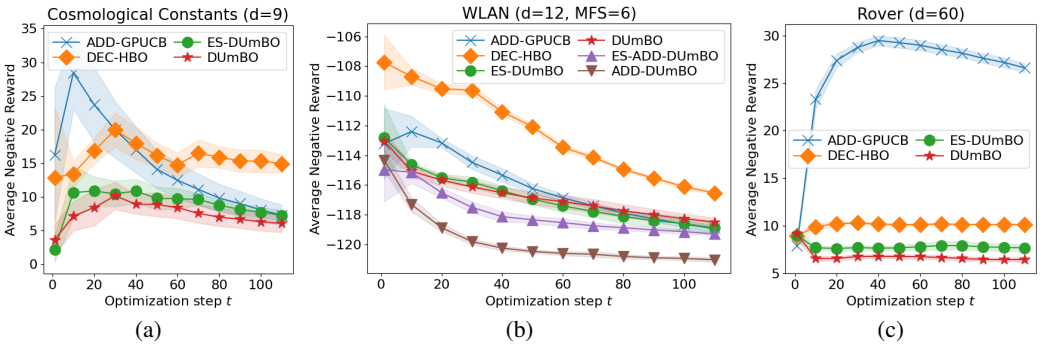


Figure 6: Performance achieved by the decomposing BO algorithms and the early-stopped versions of DuMBO for (a) the cosmological constants fine-tuning, (b) the maximization of the Shannon capacity in a WLAN and (c) the trajectory planning of a rover. The shaded areas indicate the standard error intervals.

J Wall-Clock Time

In this section, we provide wall-clock time measurements (excluding the evaluation time of the objective function) of the described BO algorithms on a synthetic function (24d Powell) and a real-world problem (WLAN) described in Appendices G and H respectively. The measurements were taken using a server equipped with two Intel(R) Xeon(R) CPU E5-2690 v4 @ 2.60GHz, with 14 cores (28 threads) each.

Figure 7 gathers the wall-clock time measurements. Observe that DuMBO does not only offer very competitive performance, it also exhibits a lower overhead when compared to the other decomposing algorithms (DEC-HBO and ADD-GPUCB). However, SAASBO and TuRBO manage to get lower

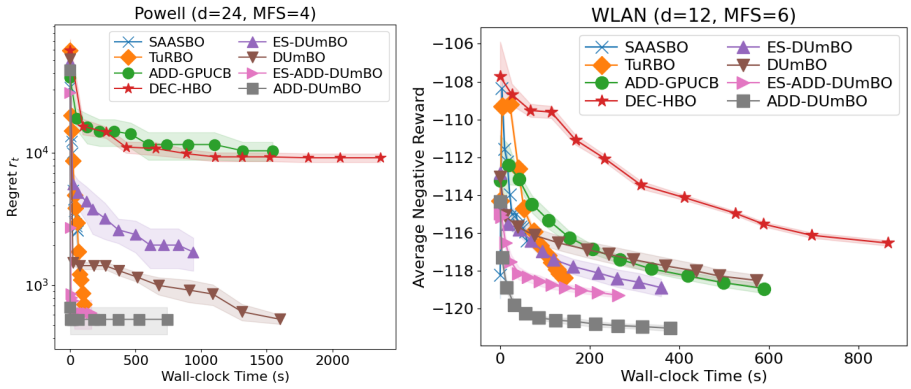


Figure 7: Performance achieved by all the described BO algorithms (including the two versions of DuMBO and their early-stopped alternatives) for (a) the 24d Powell synthetic function and (b) the maximization of the Shannon capacity in a WLAN. The shaded areas indicate the standard error intervals.

runtimes than DuMBO. This is not surprising since, by design, these methods have minimal overheads, at the expense of any theoretical guarantees.

Nevertheless, observe that the early-stopped version of DuMBO, ES-DuMBO, also reaches very good performance, with a significantly reduced response time. With ADD-DuMBO, observe that having access to the true additive decomposition of the function also reduces the overhead of the solution, since the decomposition does not need to be inferred anymore. Finally, observe that ES-ADD-DuMBO, the early-stopped version of DuMBO when the additive decomposition is provided, obtains similar results as TuRBO and SAASBO, with only a slightly larger runtime, especially on the Powell synthetic function (Figure 7(a)). Therefore, we argue that DuMBO, due to its excellent empirical performance and its satisfying runtime (thanks to its capacity to be early-stopped) is a competitive solution even in critical applications where the response time needs to be low.