Bayesian multi-objective optimization for stochastic simulators: an extension of the Pareto Active Learning method

Bruno Barracosa, Julien Bect, Héloïse Dutrieux Baraffe, Juliette Morin, Josselin Fournel, Emmanuel Vazquez

▶ **To cite this version:**

Bruno Barracosa, Julien Bect, Héloïse Dutrieux Baraffe, Juliette Morin, Josselin Fournel, et al.. Bayesian multi-objective optimization for stochastic simulators: an extension of the Pareto Active Learning method. 2022. hal-03714535v2

HAL Id: hal-03714535

https://centralesupelec.hal.science/hal-03714535v2

Preprint submitted on 19 Jul 2022

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

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

Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License

Bayesian multi-objective optimization for stochastic simulators: an extension of the Pareto Active Learning method

Bruno Barracosa[a,b], Julien Bect[b], Héloïse Dutrieux Baraffe[a], Juliette Morin[a], Josselin Fournel[a], Emmanuel Vazquez[b,*]

[a] *EDF, Research and Development, Power Systems and Energy Markets, Palaiseau, France*
[b] *Université Paris-Saclay, CNRS, CentraleSupélec, Laboratoire des signaux et systèmes, 91190, Gif-sur-Yvette, France.*

Abstract

This article focuses on the multi-objective optimization of stochastic simulators with high output variance, where the input space is finite and the objective functions are expensive to evaluate. We rely on Bayesian optimization algorithms, which use probabilistic models to make predictions about the functions to be optimized. The proposed approach is an extension of the Pareto Active Learning (PAL) algorithm for the estimation of Pareto-optimal solutions that makes it suitable for the stochastic setting. We named it Pareto Active Learning for Stochastic Simulators (PALS). The performance of PALS is assessed through numerical experiments over a set of bi-dimensional, bi-objective test problems. PALS exhibits superior performance when compared to a selection of alternative approaches based on scalarization or random-search.

Keywords: Multi-objective simulation optimization, Bayesian optimization, Stochastic simulator, Gaussian process modeling, Kriging

*Corresponding author
Email address: emmanuel.vazquez@centralesupelec.fr (Emmanuel Vazquez)

1. Introduction

We consider the problem of multi-objective optimization with stochastic evaluations, also known as multi-objective simulation optimization (Hunter et al., 2019). More precisely, let $f_1, \dots f_q$ be q real-valued objective functions defined on a search domain $\mathbb{X} \subset \mathbb{R}^d$, and assume that these objectives can only be observed with some random errors: each evaluation of the objectives at a given point $x \in \mathbb{X}$ in the search domain yields random responses $Z_j = f_j(x) + \varepsilon_j$, $1 \leq j \leq q$, where the ε_js are zero-mean random variables (the distribution of which may depend on x), mutually independent and independent from past evaluations. Our objective is to estimate the Pareto-optimal solutions of the problem:

$$\min f_1, \dots, f_q \,. \tag{1}$$

This setting has many applications in the industry when stochastic simulators are used to make decisions about uncertain systems with several, often conflicting, criteria. Examples can be found in all areas of engineering and science, for instance plant breeding (Hunter & McClosky, 2016), aircraft maintenance (Mattila & Virtanen, 2014) or electric network planning (Dutrieux et al., 2015b).

In this work, we choose to focus on Bayesian optimization algorithms, which use probabilistic models to make predictions about the functions to be optimized. One of the classical algorithms for Bayesian *deterministic* multi-objective optimization is the ParEGO algorithm (Knowles, 2006). It relies on a scalarization approach to extend the very popular single-objective Efficient Global Optimization (EGO) algorithm (Jones et al., 1998), based on the Expected Improvement (EI) criterion. Another scalarization approach, called multi-attribute Knowledge Gradient (Astudillo & Frazier, 2017), uses the Knowledge Gradient (KG) criterion instead of the EI criterion. Several Bayesian multi-objective optimization methods that do not rely on scalarization have been proposed as well. These are, for instance, the well-known Expected Hypervolume Improvement (EHVI) algorithms (Emmerich et al., 2006; Feliot et al., 2017), the Stepwise Uncertainty Reduction (SUR) approach suggested in Picheny (2015), the Predictive Entropy Search for Multi-objective Bayesian Optimization (PESMO) algorithm (Hernández-Lobato et al., 2016), the Max-value Entropy Search for Multi-objective Bayesian Optimization (MESMO) (Belakaria et al., 2020), and the Pareto Active Learning (PAL) algorithm (Zuluaga et al., 2013).

In this article, we choose to focus on PAL because it is easy to implement and inexpensive in terms of computational time, contrarily to other Bayesian approaches based on computationally-intensive criteria for point selection. The main idea of PAL consists in using confidence intervals from the posterior distribution of a Gaussian process to balance exploration and exploitation, in a way inspired by the Gaussian Process Upper Confidence Bound rule (GP-UCB) (Srinivas et al., 2010) for single-objective optimization. Zuluaga et al. (2013) present PAL as a multi-objective Bayesian optimization tool directed at cases

where "evaluating the objective function is expensive and noisy". However, as will be discussed in this article, PAL is fundamentally an algorithm designed for the optimization of deterministic objective functions over a finite input space, and significant limitations arise when considering truly stochastic (random) evaluations.

This article brings two contributions: (1) an extension of the PAL algorithm, called PALS, making it suitable for stochastic evaluations, and (2) a numerical benchmark comparing the performance of the proposed approach with several competing approaches based on scalarization and random-search.

The article is organized as follows. Section 2 introduces in more detail the problem of multi-objective optimization for stochastic simulators. Section 3 describes the original PAL algorithm. Section 4 presents the proposed extension for stochastic simulators. Section 5 details the numerical experiments and their main results. Section 6 draws conclusions and suggests several perspectives. A separate supplementary material file is available.

2. Bayesian multi-objective optimization for stochastic simulators

Consider a black-box stochastic simulator whose outputs are noisy evaluations of q latent functions $f_1, \ldots, f_q : \mathbb{X} \mapsto \mathbb{R}$ over a search space $\mathbb{X} \subset \mathbb{R}^d$, $d \in \mathbb{N}$. The vector of function values $(f_1(x), \ldots, f_q(x))$ will be denoted by $\boldsymbol{f}(x)$. Suppose that the functions $f_1, \ldots, f_q$ represent (possibly conflicting) costs that we want to minimize.

In a multi-objective optimization problem, the goal is to identify the solutions that represent the best possible trade-offs among the q objectives. This is usually defined using the Pareto domination rule: for any two points x, $x' \in \mathbb{R}^d$ with images $z = (z_1, \ldots, z_q) = \boldsymbol{f}(x)$, $z' = (z'_1, \ldots, z'_q) = \boldsymbol{f}(x') \in \mathbb{R}^q$, we write $z \prec z'$ (z dominates z' in the sense of Pareto), when $z_j \leq z'_j$ for all j, with at least one of the inequalities being strict. The set $\mathcal{P}$ of all non-dominated points is called the *Pareto set*:

$$\mathcal{P} = \{x \in \mathbb{X} : \nexists x' \in \mathbb{X},\ \boldsymbol{f}(x') \prec \boldsymbol{f}(x)\}\,. \tag{2}$$

The *Pareto front* $\mathcal{F} \subset \mathbb{R}^q$ is by definition the image $\boldsymbol{f}(\mathcal{P})$ of the Pareto set by the objective functions. Figure 1 illustrates the Pareto front of a finite set of evaluations of two objective functions.

The goal of multi-objective optimization is to obtain a good approximation $\widehat{\mathcal{P}}_n$ of $\mathcal{P}$ using a sequence $(X_1, \ldots, X_n)$, $n \geq 1$, of evaluation points and the corresponding evaluation results. (The use of capital letters to denote the evaluation points indicates that these points depend on past evaluation results.)

In this article, we assume a stochastic setting, where evaluations are corrupted by additive, normally distributed, homoscedastic noise. In other words, instead of

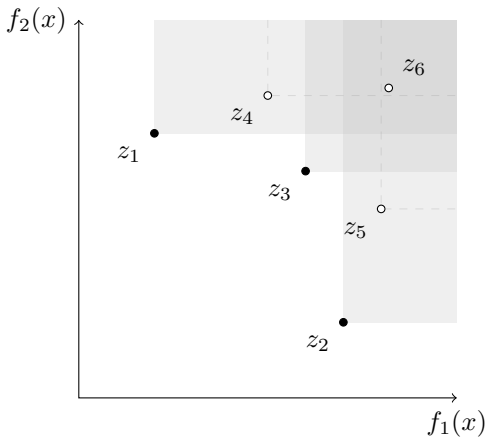

Figure 1: Example of a Pareto front $\mathcal{F} = \{z_1, z_2, z_3\}$, in a bi-objective example with finite search domain $\mathbb{X} = \{x_1, \ldots, x_6\}$.

observing the values of $f_1, \ldots, f_q$ at $X_1, \ldots, X_n$, we observe random responses

$$Z_{i,j} = f_j(X_i) + \varepsilon_{i,j}, \quad 1 \leq i \leq n,\ 1 \leq j \leq q,$$

where the random variables $\varepsilon_{i,j}$ are mutually independent, with Gaussian distributions $\mathcal{N}(0, \sigma_j^2)$ for some unknown $\sigma_j^2 > 0$.

Using a Bayesian framework provides a practical solution to handle this stochastic setting. The main idea is to use a prior probability distribution, often simply called the prior, to model each objective f_j as a sample path of a random process ξ_j. For convenience, we assume moreover that the ξ_js are independent Gaussian processes, since it makes it possible to easily obtain the posterior distributions of the models ξ_j conditional on the observations $Z_{i,j}$. In this case indeed, the ξ_js remain independent and Gaussian a posteriori (i.e., given the evaluation results). We denote by $\mu_{n,j}$ and $k_{n,j}$ their posterior mean and covariance functions, which can be derived by solving systems of linear equations (see, e.g. Chiles & Delfiner, 1999; Rasmussen & Williams, 2006).

Posterior distributions based on past observations can be used simultaneously to choose additional evaluation points, and to generate predictions of the Pareto front. Pareto front predictions can be generated through conditional simulations (Figure 2), making it possible not only to estimate the Pareto front, but also to quantify the remaining uncertainty (Binois et al., 2015).

As pointed out in the introduction, several Bayesian multi-objective optimization algorithms have been proposed in the literature, but few of them can actually be used under a stochastic framework. Astudillo & Frazier (2017) propose an algorithm that takes as input a user preference and converges to a particular solution of the Pareto front (i.e., does not provide an estimate of the entire

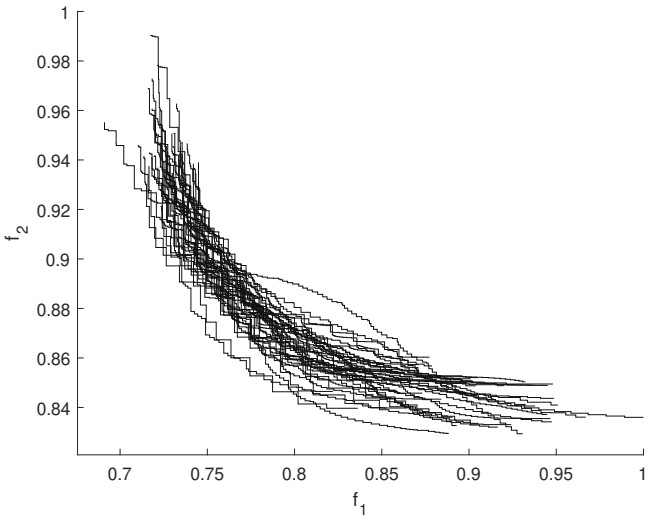

Figure 2: Bi-objective example of normalized random realizations of Pareto fronts constructed from probabilistic models.

Pareto set and front). The PESMO algorithm (Hernández-Lobato et al., 2016) on the other hand, which has been proposed in the machine learning literature, can be used to jointly estimate the Pareto set and front, as considered in this article. However, this algorithm has a high implementation complexity and a non-negligible execution cost—it relies at each iteration on conditional simulations of the GP models, a genetic algorithm and an expectation-propagation algorithm.

Unlike PESMO, the PAL algorithm proposed by Zuluaga et al. (2013) is a simple algorithm, but it is not as such suitable for the stochastic case. In the next sections, we introduce the original PAL algorithm, discuss its limitations, and propose modifications to deal with the stochastic case.

Remark 1. *It is worth noting that several algorithms from the literature of budget allocation / ranking and selection, namely MOCBA (Lee et al., 2010), SCORE (Pasupathy et al., 2014), and M-MOBA (Branke & Zhang, 2015) algorithms, use similar ideas to those outlined above, but with one notable difference: they assume no correlation between the values of the objective functions at distinct points in the search space. Recently, an algorithm using a mix of Bayesian optimization and ranking and selection approaches was proposed by Rojas Gonzalez et al. (2020) and named SK-MOCBA.*

3. The original PAL algorithm

In the following, we assume that $\mathbb{X} \subset \mathbb{R}^d$ is a (possibly large) finite set, corresponding for instance to a discrete grid or a large random sample from a uniform

distribution on a bounded set. The PAL algorithm builds a sequence $(X_n)_{n\geq 1}$ of evaluation points using, at each iteration, a classification of all the points $x \in \mathbb{X}$ of the search space as potentially Pareto-optimal or not, based on the posterior distributions of the Gaussian processes ξ_j. More precisely, the algorithm partitions $\mathbb{X}$ into three sets at iteration n: a set P_n of points that are deemed Pareto-optimal, a set N_n of points that are deemed dominated, and the set $U_n = \mathbb{X} \setminus (P_n \cup N_n)$ of points that remain unclassified. Then, the PAL algorithm improves the approximation of the Pareto set by choosing an additional evaluation point in $P_n \cup U_n$, where uncertainty is maximal (in a sense defined below).

To build P_n, U_n and N_n, each x of the search space is associated to a region $R_n(x) \subset \mathbb{R}^q$, $n \geq 0$, in the objective space, which quantifies the uncertainty about the values of the objectives at that point. For each $n \geq 0$, the region $R_n(x)$ is constructed as follows. First, for each $x \in \mathbb{X}$ and each $j \in \{1, \ldots, q\}$, compute the posterior means $\mu_{n,j}(x)$ and posterior variances $\sigma^2_{n,j}(x) = k_{n,j}(x,x)$ of the Gaussian random variables $\xi_j(x)$ modeling the unknown values of the f_js at x, given observations $Z_{1,j}, \ldots, Z_{n,j}$ (for $n = 0$, they are equal to the prior means and variances). Then, define hyper-rectangles $Q_n(x) \subset \mathbb{R}^q$ capturing the uncertainty of the predictions, as illustrated in Figure 3:

$$Q_n(x) = \left\{ z \in \mathbb{R}^q : \boldsymbol{\mu_n}(x) - \beta^{1/2}\boldsymbol{\sigma_n}(x) \prec z \prec \boldsymbol{\mu_n}(x) + \beta^{1/2}\boldsymbol{\sigma_n}(x) \right\}, \qquad (3)$$

where $\boldsymbol{\mu_n}(x) = (\mu_{n,1}(x), \ldots, \mu_{n,q}(x))$, $\boldsymbol{\sigma_n}(x) = (\sigma_{n,1}(x), \ldots, \sigma_{n,q}(x))$, and $\beta > 0$ is a positive scaling factor. Finally, for each $x \in \mathbb{X}$, the regions used for classification are defined as

$$R_n(x) = \begin{cases} \{\mu_n(x)\} & \text{if } x \in \{X_1, \ldots, X_n\}, \\ R_{n-1}(x) \cap Q_n(x) & \text{otherwise,} \end{cases} \qquad (4)$$

with the convention that $R_{-1}(x) = \mathbb{R}^q$. (NB: in the original article by Zuluaga et al. (2013), a point $x \in \mathbb{X}$ already visited is artificially assigned a zero posterior variance; here, we prefer to state that the corresponding region $R_n(x)$ collapses to $\mu_n(x)$, which is equivalent.)

For each $R_n(x)$, define an optimistic outcome $R_n^{\min}(x)$ and a pessimistic outcome $R_n^{\max}(x)$ as the best possible outcome and the worst possible outcome within $R_n(x)$ according to the Pareto domination rule (see Figure 4). Then, each point $x \in \mathbb{X}$ is classified in P_n, U_n or N_n according to the following rules illustrated in Figure 5:

$$P_n = \left\{ x \in \mathbb{X} \;\middle|\; \nexists x' \in \mathbb{X},\ R_n^{\min}(x') + \epsilon \prec R_n^{\max}(x) - \epsilon \right\}, \qquad (5)$$

where $\epsilon \in \mathbb{R}^q_+$ is a vector of margin parameters controlling the acceptation in P_n and N_n. In other words, $x \in P_n$ when the (ϵ–shifted) pessimistic outcome $R_n^{\max}(x)$ is not dominated by the (ϵ–shifted) optimistic outcome $R_n^{\min}(x')$

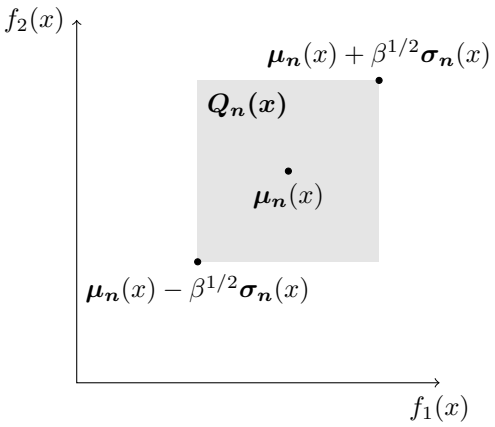

Figure 3: Bi-objective example of a prediction uncertainty region $Q_n(x)$.

of any other point x'. Conversely,

$$N_n = \left\{ x \in \mathbb{X} \;\middle|\; \exists x' \in \mathbb{X} \setminus \{x\},\ R_n^{\max}(x') - \epsilon \prec R_n^{\min}(x) + \epsilon \right\}, \tag{6}$$

or in other words, $x \in N_n$ when the (ϵ–shifted) optimistic outcome $R_n^{\min}(x)$ is dominated by at least one of the the (ϵ–shifted) pessimistic outcome $R_n^{\max}(x')$ of another point x'; and finally $U_n = \mathbb{X} \setminus (P_n \cup N_n)$.

According to Zuluaga et al. (2013), the margin parameter ϵ provides flexibility around the uncertainty region, and can be defined globally or individually for each objective.

The parameter β plays the role of a scaling factor for the uncertainty regions and is defined at each iteration. Zuluaga et al. (2013) suggest using an increasing sequence β_n defined at each iteration n as

$$\beta_n = 2\log(q|\mathbb{X}|\pi^2 n^2/6\delta), \tag{7}$$

with $\delta \in [0, 1]$. The proposed approach allows to ensure an upper bound of the hypervolume error with probability $1 - \delta$, by specifying δ. Actual convergence and upper bound on the maximal number of needed iterations are presented for the case where intersecting regions are used, and a squared exponential kernel is adopted with parameters that are not re-estimated at each iteration (refer to Corollary 2 in the original paper for details).

The last step of the algorithm consists in selecting a new evaluation point X_{n+1} among the non-visited points in the union $P_n \cup U_n$ for which uncertainty is maximal, in the sense of the Euclidean distance between the pessimistic and the

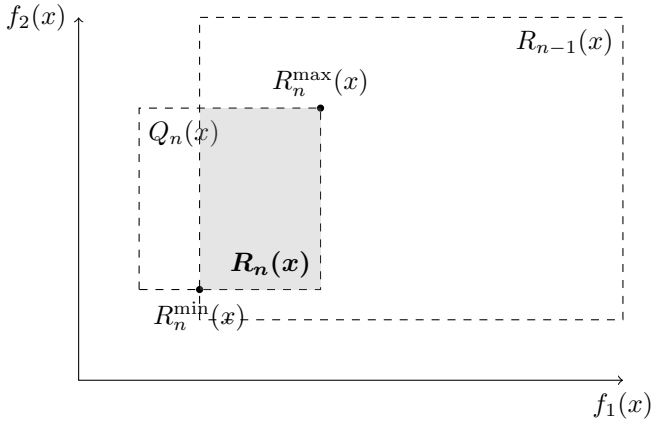

Figure 4: Bi-objective example of the construction of a classification region $R_n(x)$ (filled region), and the respective optimistic $R_n^{\min}(x)$ and pessimistic $R_n^{\max}(x)$ outcomes.

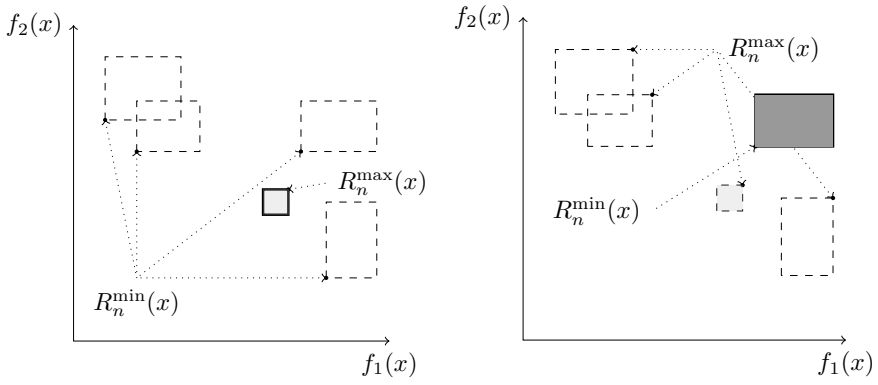

Figure 5: Illustration of the classification of a Pareto-optimal point (left) presented in light gray, and the classification of a non-Pareto-optimal point (right) presented in dark gray, while unclassified points remain as non-filled square dashed outlines.

optimistic outcomes (i.e., the diameter of $R_n(x)$):

$$X_{n+1} = \underset{x \in (U_n \cup P_n) \setminus S_n}{\operatorname{argmax}} \left\| R_n^{\min}(x) - R_n^{\max}(x) \right\|_2 , \tag{8}$$

where S_n denotes the set of visited points.

Algorithm 1 summarizes the PAL algorithm.

Remark 2. *It appears from the right-hand side of (8) that PAL expects the objectives to be suitably normalized, in order to be comparable. Indeed, the criterion at any given point involves a sum of differences of objective values:*

$$\left\| R_n^{\min}(x) - R_n^{\max}(x) \right\|_2^2 = \sum_{j=1}^{q} \left(R_{n,j}^{\min}(x) - R_{n,j}^{\max}(x) \right)^2 .$$

We will assume for simplicity that such a normalization has been made beforehand. When this is not possible, the range of the objective functions can be learned on-the-fly as done, e.g., by Feliot et al. (2017).

Remark 3. *In practice, in Bayesian optimization, it is advisable to start the optimization procedure using a set of n_0 preliminary evaluations, which make it possible to select the parameters of the Gaussian process model before sequential selection of points. This initial set of results is called "training set" by Zuluaga et al. (2013), or more traditionally initial design of experiments. In this case, the index n in Algorithm 1 corresponds to the number of simulations performed after the initial design, and the posterior means and variances at iteration n are in fact computed with respect to $n + n_0$ evaluation results.*

4. PALS: modification of the PAL algorithm for stochastic simulators

The original PAL algorithm presents limitations that make it unsuitable for stochastic evaluations. To circumvent these limitations, we propose several modifications, which can in fact be seen as *simplifications* of the original algorithm. The simplifications, together with an extension to batch evaluations, are detailed below. As in PAL, it is assumed that $\mathbb{X}$ is a finite set. The resulting algorithm, summarized in Algorithm 2, is called PALS (S for Stochastic).

Modification 1: Taking into account prediction variance at visited points. By artificially setting the prediction variance at visited points to zero (equivalently, by collapsing $R_n(x)$ to $\mu_n(x)$ as in (4)), Zuluaga et al. (2013) prevent the PAL algorithm from visiting any point of the search space more than once. This behavior is also enforced by the explicit requirement (see (8)) that X_{n+1} should belong to the set $(P_n \cup U_n) \setminus S_n$, where S_n denotes the set of visited points. In fact, this approach can be seen as using a "nugget" component (Matheron, 1963) in the GP prior, which would be appropriate for a very irregular deterministic

Algorithm 1: The original PAL algorithm

Input : input space $\mathbb{X}$; GP priors; ϵ_q; β_n
Output : predicted Pareto set $\widehat{\mathcal{P}}$
$P_{-1} = \varnothing,\ N_{-1} = \varnothing,\ U_{-1} = \mathbb{X},\ S_{-1} = \varnothing$
$R_{-1}(x) = \mathbb{R}^q$ for all $x \in \mathbb{X}$
$n = 0$, all_classified = false
repeat
  Obtain $\boldsymbol{\mu}_n(x)$ and $\boldsymbol{\sigma}_n(x)$ for all $x \in \mathbb{X}$,
   $\{\boldsymbol{\mu}_n(X_i) = (Z_{i,1}, \ldots, Z_{i,q})$ `and` $\boldsymbol{\sigma}_n(X_i) = 0, \forall X_i \in S_n\}$
  $R_n(x) = R_{n-1}(x) \cap Q_{\mu_n, \sigma_n, \beta_n}(x)$ for all $x \in \mathbb{X}$
  $P_n = P_{n-1}$, $N_n = N_{n-1}$, $U_n = \varnothing$
  forall $x \in U_{n-1}$ **do**
    if $\left(\nexists x' \in \mathbb{X} \setminus \{x\} : R_n^{\min}(x') + \epsilon \prec R_n^{\max}(x) - \epsilon\right)$ **then**
      $P_n = P_n \cup \{x\}$
    else if $\left(\exists x' \in \mathbb{X} \setminus \{x\} : R_i^{\max}(x') - \epsilon \prec R_i^{\min}(x) + \epsilon\right)$ **then**
      $N_n = N_n \cup \{x\}$
    else
      $U_n = U_n \cup \{x\}$
  if $U_n = \varnothing$ **then**
    all_classified = true
  else
    Choose $X_{n+1} = \operatorname{argmax}_{x \in (U_n \cup P_n) \setminus S_n} \left\| R_n^{\min}(x) - R_n^{\max}(x) \right\|_2$
    $S_{n+1} = S_n \cup \{X_{n+1}\}$
    Sample $Z_{n+1,j} = f_j(X_{n+1}) + \varepsilon_{n+1,j}$ for all $j \in \{1, \ldots, q\}$
    $n = n + 1$
until all_classified;
return $\widehat{\mathcal{P}} = P_n$

simulator, but is not for a stochastic simulator—where replicated simulations at a given point yield independent and identically distributed random responses. To circumvent this limitation, we remove the requirement that X_{n+1} should belong to $\mathbb{X} \setminus S_n$ and use the correct posterior variance at all points, visited or not; more explicitly, we replace (4) and (8) with

$$R_n(x) = R_{n-1}(x) \cap Q_n(x)\,. \tag{9}$$

and

$$X_{n+1} = \underset{x \in U_n \cup P_n}{\operatorname{argmax}} \left\| R_n^{\min}(x) - R_n^{\max}(x) \right\|_2 . \tag{10}$$

As a consequence, PALS can visit a given point several times and thus the number of iterations of the algorithm is not bounded as in PAL by the cardinality of the search domain. We introduce a user-defined maximum budget of evaluations $n_{\max}$, which will constitute our main stopping criterion (since it is often extremely costly, in the problems that we consider, to reach a situation where all the points are classified in P_n or in N_n). We note, however, that keeping the cardinality of U_n as an additional stopping criterion is the only reason why the distinction between U_n and P_n is still relevant.

Modification 2: Removing the intersection of uncertainty regions. Zuluaga et al. (2013) define R_n at each iteration as the intersection of R_{n-1} and Q_n in (4) in order to get a non-increasing sequences of regions—a monotonicity property which plays a role in the proof of their theoretical results. However, problematic situations can arise as a result of this definition, where $Q_n(x)$ is not contained in $R_{n-1}(x)$ for some x at some iteration n. When this happens, $R_n(x)$ is empty and the subsequent behavior of the algorithm is not properly defined since the pessimistic and optimistic outcomes at x, $R_n^{\max}(x)$ and $R_n^{\min}(x)$, are not defined. In the PALS algorithm we remove intersections to address this issue, and thus further simplify (9) to

$$R_n(x) = Q_n(x). \tag{11}$$

As a consequence, the subsets P_n and N_n defined by (5)–(6) loose their monotonicity property as well, and thus membership of all the points in $\mathbb{X}$ to P_n, N_n or U_n must be re-evaluated at each iteration.

Remark 4. *As an alternative, it is possible to keep the original idea of using intersections and simply "correct" the sets $R_n(x)$ so that $R_n(x)$ always contains $\mu_n(x)$. This can be achieved by taking $R_n(x)$ to be the smallest hyper-rectangle that contains both $R_{n-1} \cap Q_n(x)$ and $\{\mu_n(x)\}$. Note that doing so artificially fixes the problem of empty regions, but nonetheless destroys the monotonicity property, which was the original motivation of Zuluaga et al. (2013) for introducing intersections. For the sake of completeness, this approach will also be assessed in our experimental section.*

Modification 3: Batches of simulations. When the output variability is large, a single simulation provides a very noisy evaluation of the quantities of interest, which are the expected responses f_j of the simulator, and therefore yields by itself little progress in the optimization procedure. A natural idea to gain more information, when parallel computing resources are available, is to perform several simulations at the same point at each iteration of the algorithm. An approach of this type has been proposed for instance by Binois et al. (2019). We denote by k the size the simulation batches, i.e., the number of simulations performed at the same point at each iteration.

The number of simulation results available after n iterations with batch size k is equal to $N = nk$, and thus grows quickly if k is large (for instance $k = 200$ or more in Section 5). Such a large number of results poses apparently a challenge for Gaussian process regression, whose computational complexity is $O(N^3)$. However, this complexity can be reduced, without any approximation, since Gaussian process regression with Gaussian noise only uses the empirical mean and variance of the simulation results at each point $x \in \mathbb{X}$ where at least one simulation was made (see, e.g., Dutrieux et al., 2015a; Binois et al., 2018): a careful implementation leveraging this fact has a complexity that scales cubically with the number of "visited" points, which does not grow with k.

Remark 5. *Another approach, not pursued in this article, would consist in sampling simultaneously multiple locations at each iteration, as proposed by Habib et al. (2016), for instance.*

Predicting the Pareto front and set. PAL uses the condition $U_n = \varnothing$ as a stopping criterion, and then estimates the unknown Pareto set using the set P_n obtained at the final iteration. With the introduction of a budget-based stopping condition, however, PALS typically stops before all points are classified. A first approach to estimate the Pareto front and set consists in applying Pareto domination rule directly to the posterior means of the GP models, in a "plug-in" manner. This approach, however, does not take the residual uncertainty into account. An alternative approach would be to generate conditional sample paths of the Gaussian processes instead to represent possible realizations of the objective functions, and then use them to estimate for each point in the objective space the probabilities of that point being dominated (also called empirical attainment function in Binois et al. (2015)), and for each point in the input space, the probabilities of belonging to the Pareto set (also called coverage probabilities). An illustration is provided for the Pareto front and set in Figure 6. It is worth noting that a high confidence about the Pareto front location does not imply a high confidence about the location of the Pareto set, and vice versa.

Although providing enriched information about the Pareto front and set, this approach entails a higher computational cost. Therefore, for the remaining sections, the plug-in approach is used to compute estimates of the Pareto front and the Pareto set (at the final iteration, but also at each iteration for the

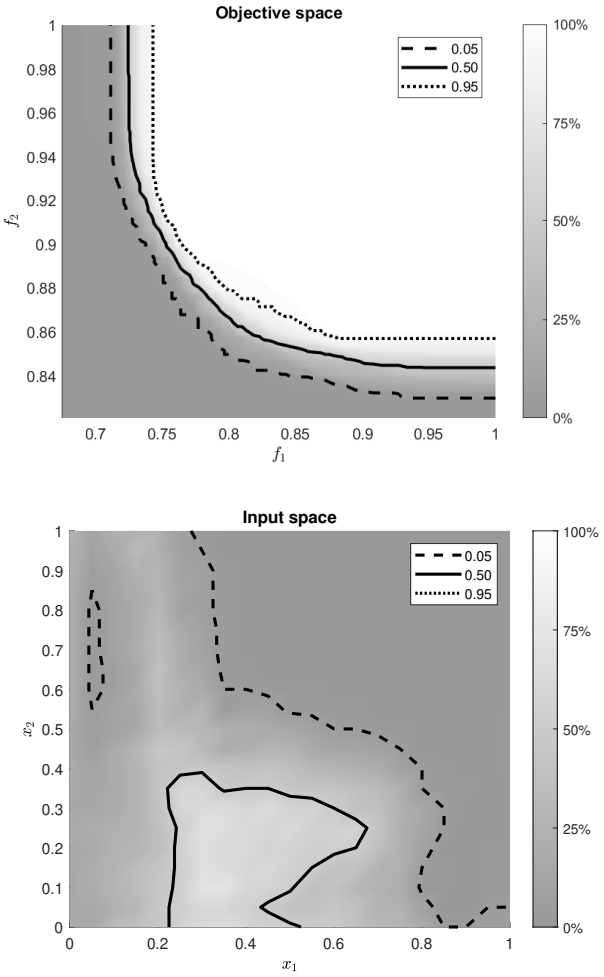

Figure 6: Bi-objective and bi-dimensional example of a normalized heat map representing the probability of a point being dominated for the objective space (top) and the probability of belonging to the Pareto set (bottom) constructed from a probabilistic model (built from the same data and model as in Figure 2, using 40 sample paths). Level curves shown for 5%, 50%, and 95% levels.

Algorithm 2: The PALS algorithm

Input : input space $\mathbb{X}$; GP priors $\mu_{0,j}, \sigma_{0,j}, k_j, \forall j$; ϵ_j; β_i; $n_{\max}$; k.
Output : predicted Pareto set $\widehat{\mathcal{P}}$.
$n = 0$, all_classified = false
repeat
  Obtain $\boldsymbol{\mu}_n(x)$ and $\boldsymbol{\sigma}_n(x)$ for all $x \in \mathbb{X}$
  $R_n(x) = Q_{\mu_n,\sigma_n,\beta_n}(x)$ for all $x \in \mathbb{X}$
  $P_n = \varnothing$, $N_n = \varnothing$, $U_n = \varnothing$
  forall $x \in \mathbb{X}$ **do**
    if $\left(\nexists x' \in \mathbb{X} \setminus \{x\} : R_n^{\min}(x') + \epsilon \prec R_n^{\max}(x) - \epsilon\right)$ **then**
      $P_n = P_n \cup \{x\}$
    else if $\left(\exists x' \in \mathbb{X} \setminus \{x\} : R_n^{\max}(x') - \epsilon \prec R_n^{\min}(x) + \epsilon\right)$ **then**
      $N_n = N_n \cup \{x\}$
    else
      $U_n = U_n \cup \{x\}$
  if $U_n = \varnothing$ **then**
    all_classified = true
  else
    Choose $X_{n+1} = \operatorname{argmax}_{x \in (U_n \cup P_n)} \left\| R_n^{\min}(x) - R_n^{\max}(x) \right\|_2$
    Obtain k samples:
    $Z_{n+1,j}^{(\ell)} = f_j(X_{n+1}) + \varepsilon_{n+1,j}^{(\ell)},\ 1 \le j \le q,\ 1 \le \ell \le k$
    $n = n + 1$
until all_classified ***or*** $n \ge n_{\max}$;
return $\widehat{\mathcal{P}}$

purpose of algorithm monitoring and, in our benchmarks, for error assessment).

5. Numerical experiments

5.1. Methodology

In this section, PALS is compared with a set of alternative optimization methods: 1) a pure random search approach (PRS), where evaluation points are drawn from the uniform probability distribution on $\mathbb{X}$; 2) an alternative random search approach, where evaluation points are drawn from a model-based probability distribution concentrated on "promising" regions; and 3) two scalarization approaches, which are modifications of the ParEGO algorithm (Knowles, 2006).

More precisely, the alternative random search approach consists in drawing the next evaluation point from a distribution proportional to the probability of that point being misclassified, according to the posterior distributions of the Gaussian processes ξ_j.

For the scalarization techniques, we consider two modified versions of the ParEGO algorithm suitable for a stochastic setting. ParEGO relies on the use of

EI as an infill criterion which is suitable only for deterministic observations. The first version, hereafter named ParEGO-EI$_m$, uses the EI$_m$ criterion defined in Vazquez et al. (2008), which considers the Expected Improvement with respect to the best prediction over the visited points (as seen used in the implementation in the PESM branch of Spearmint (Spearmint, 2016)). The second one, hereafter named ParEGO-KG, uses the Knowledge Gradient (KG) sampling criterion instead, which is suitable for a stochastic setting (Frazier et al., 2009).

For the PALS approach, when an increasing β is considered (cf. Equation (7)), the parameter δ is taken equal to 0.05 as in the original article. We consider $\epsilon = 0$ for all experiments, given that this parameter has not shown a significant effect in our numerical experiments (details available in the supplementary material).

Numerical experiments are performed over a set of nine bi-objective and bi-dimensional test problems subject to additive homoscedastic Gaussian noise and defined over a finite 21×21 input space (see Appendix A). A total of 200 optimization runs are performed for each problem and optimization method.

All the methods except PRS rely on GP modeling: the GP priors on the two objectives are independent, with a constant unknown mean and a Matérn 5/2 covariance function. The unknown mean is integrated out using an improper uniform prior ("ordinary kriging"), and the parameters of the covariance function are estimated at each iteration by the Restricted Maximum Likelihood (ReML) method (see, e.g., Santner et al., 2003).

Parameters are initialized based on an initial design which is built by choosing among 1000 initializations of 20 randomly selected points so as to maximize the minimal distance among them. The initial observations are obtained by performing 10 evaluations at each initial point. A total budget $n_{\max}$ of 50000 evaluations is allowed at each experiment in addition to the initial $20 \times 10 = 200$ evaluations. At each iteration, a batch of k evaluations is performed at the selected point. The Pareto set and front predictions are generated, for all methods, using the plug-in approach described in Section 4.

The performance of the methods is assessed using two metrics, averaged over the 200 runs: 1) the volume of the symmetric difference between the predicted dominated region and the true dominated region, and 2) the misclassification rate, meaning the percentage of points incorrectly classified in the Pareto set prediction. See Appendix B for details. These metrics provide quantitative information on the error between the prediction and the correct Pareto set or front.

Unless otherwise mentioned, the setup used for experiments is as follows: no intersection considered, a constant value for β corresponding to the 0.50 probability interval, a null value for ϵ, and a value of 200 for k.

5.2. Results

Each of the numerical experiments presented in this section has been carried out on the nine test problems. For brevity, only some representative results are presented; the remaining results are available in the supplementary material.

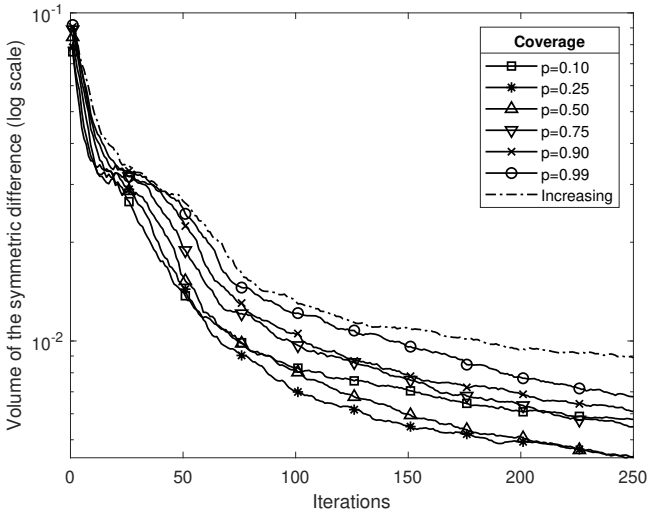

Figure 7: Comparison of average volume of the symmetric difference for problem g_2, between the increasing-β approach and fixed β for coverage values of 0.10, 0.25, 0.50, 0.75, 0.90, and 0.99.

Influence of the parameter β of PALS. We are interested in testing the usefulness of an increasing β value for the algorithm's performance. For this purpose, several constant β values were chosen to generate intervals with different coverage probabilities p. More precisely, in this context, we define the coverage probability as the probability that, for any particular objective j, at any point $x \in \mathbb{X}$, the function value lies within the region $\mu_j(x) \pm \beta^{1/2}\sigma_j(x)$. Since the posterior is Gaussian, this definition implies that $\sqrt{\beta} = \Phi^{-1}(0.5+0.5p)$, where Φ denotes the standard normal cumulative distribution function. These values are compared with the increasing-β approach proposed in the original PAL algorithm (cf. Equation (7)).

The choice of β impacts the size of the uncertainty region around the prediction. A small β allows the algorithm to focus only on the most promising regions, by quickly excluding those that seem less interesting. On the other hand, a large β favors exploration by taking longer to exclude points from being visited.

In terms of algorithm performance, an example is presented in Figure 7 for problem g_2, for the volume of the symmetric difference error metric, comparing the increasing-β approach with fixed β for coverage values of 0.10, 0.25, 0.50, 0.75, 0.90, and 0.99.

When looking at the other problem results (as presented in the supplementary material), the choice of the ideal β seems to be problem-dependent and varies depending on whether the metric used focuses on the quality of Pareto front or Pareto set predictions. However, constant values corresponding to 0.50 and 0.75 probability regions have shown an overall interesting performance.

Additionally, experiments were performed to assess the impact of the noise level on the choice of the ideal β. Figure 8 presents the same problem as Figure 7 but with two different noise levels. In our benchmark, reduced noise levels seem to render the β choice less important given the similar performance observed among choices. Small coverage levels (10% or 25%), although promising to accelerate classification, should be avoided given the low performance when considering the Pareto front prediction quality metric. Coverage levels between 50% and 90% have shown satisfying results across a multitude of problems and noise levels for both metrics. The increasing-β approach (in the form proposed by the PAL authors) brings no significant advantage when compared to a fixed β for the classification error metric, but usually underperforms for the volume of the symmetric difference metric.

Considering the results obtained, a β with constant value corresponding to the 0.50 probability interval is used in PALS in the following experiments.

About intersections. In the PALS algorithm, intersecting hyper-regions does not seem to accelerate or improve the results. Figure 9 presents a comparison between the non-intersecting and intersecting variants of PALS by showing for the different problems the average performance metrics of using (or not) intersections, normalized by the reference PRS performance. It turns out that average results in the middle and final iterations are very similar, with a slightly better performance of PALS without intersection for the misclassification rate metric. Detailed results can be consulted in supplementary material. Given these results, only the non-intersecting version of PALS will be considered for the remainder of the article.

Influence of the batch size. When using a large budget of evaluations, the results show no significant impact of the batch size k in the average metrics obtained at the final iteration. However, it is worth noting that a smaller batch size for most problems allowed a faster reduction in metric value at initial iterations. Figure 10 depicts the comparison between the average metrics for problem g_5 for k values of 200, 500, 1000 and 2000. This acceleration of the initial convergence can also be observed for the other problems on the plots available in the supplementary material. Given the results, $k = 200$ will be used for the remainder of the article.

Comparison with other approaches. This section compares the results of the different approaches: PRS, an alternative random search approach (we will name it "Concentrated Random Sampling" (CoRS)), two modified ParEGO approaches, and PALS. For PALS, the default parameters are used: a β with constant value corresponding to the 0.50 probability interval, an ϵ of 0, no intersections, and a k value of 200.

Figure 11 presents the evolution of the performance metrics on problem g_8 for the considered methods. Performance illustration for the remaining problems is available in the supplementary material. We can observe that PALS presents

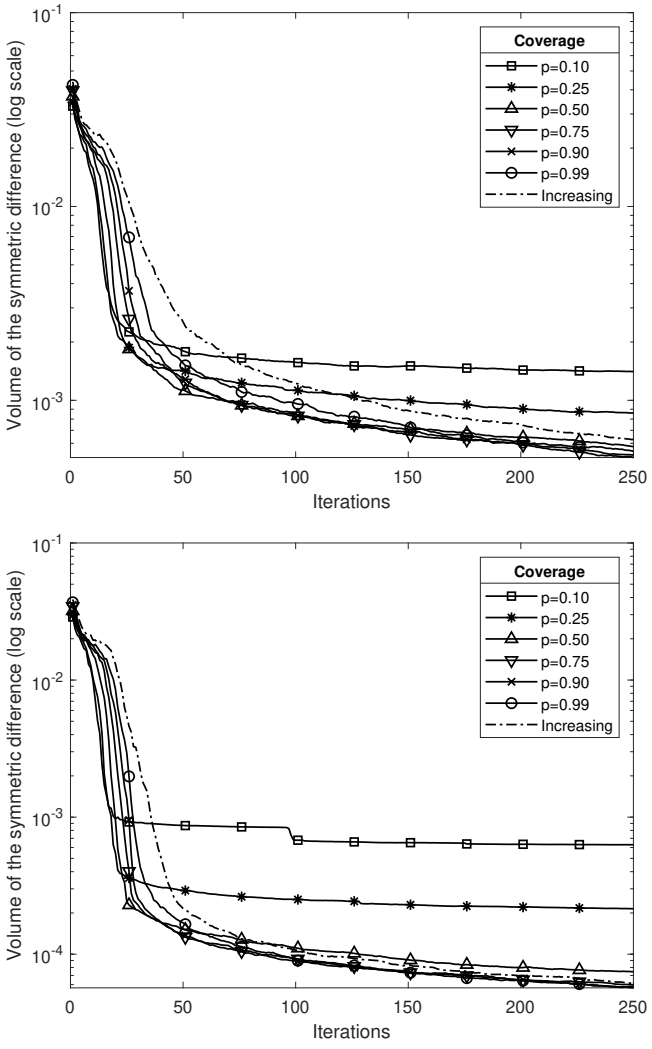

Figure 8: Average volume of the symmetric difference for problem g_2 for different β values, with the problem noise standard deviation divided by 10 (top) and by 100 (bottom)

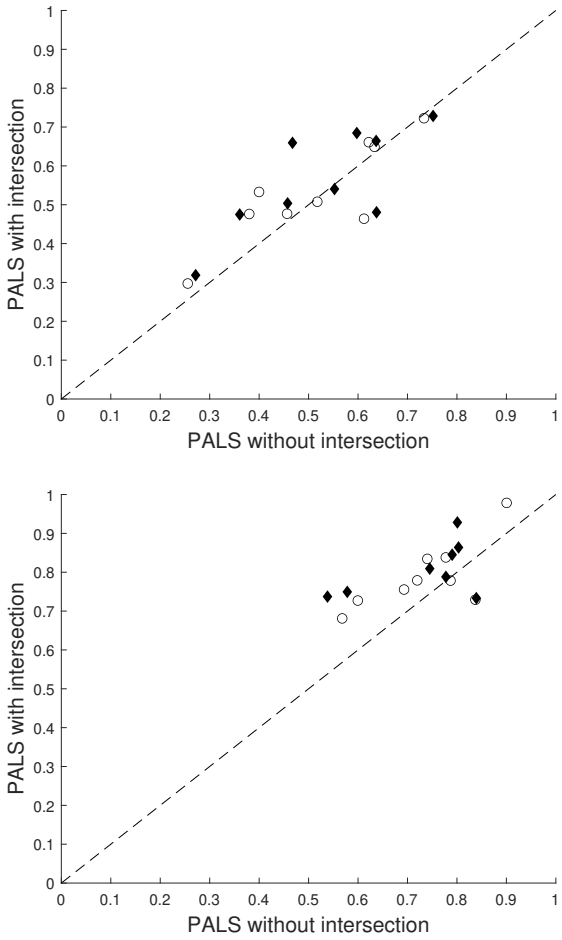

Figure 9: Performance comparison between PALS algorithm with and without intersection, normalized by the reference PRS performance. Average performance, for each problem for the volume of the symmetric difference (top) and misclassification rate (bottom), considering 200 random initialization, obtained after a budget of 25000 evaluations (circles) and 50000 evaluations (diamonds).

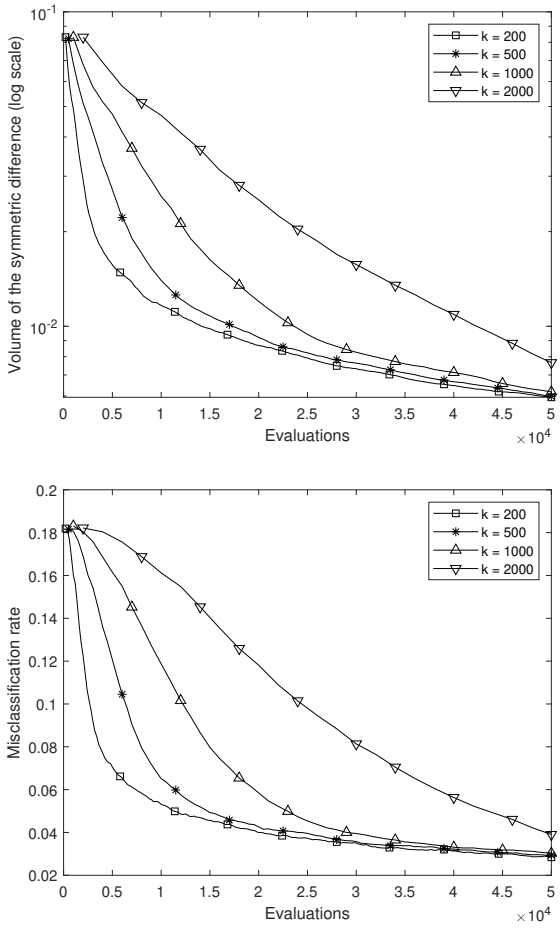

Figure 10: Average performance metrics volume of the symmetric difference (top) and misclassification rate (bottom) for problem g_5, when considering k of 200, 500, 1000 and 2000.

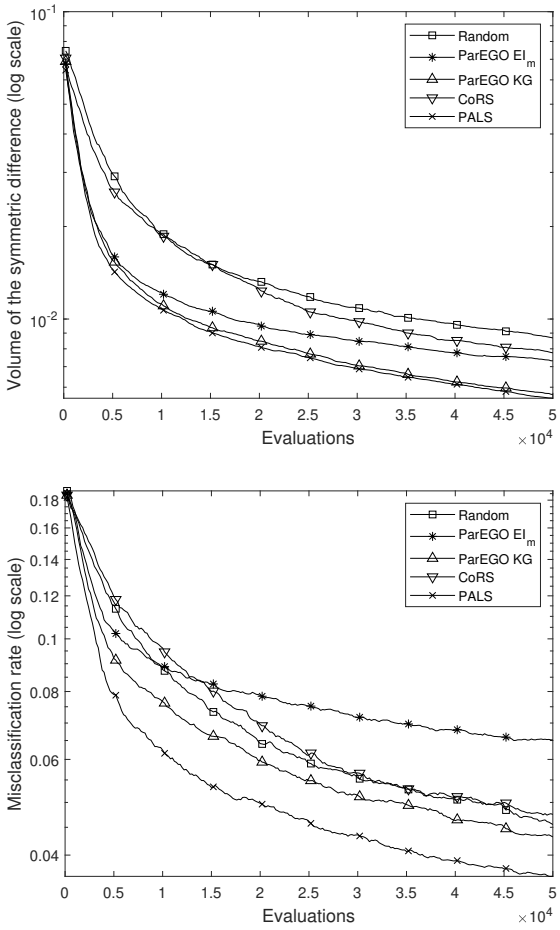

Figure 11: Average volume of the symmetric difference (top) and misclassification rate (bottom) on test problem g_8 for PRS, CoRS, ParEGO-EI$_m$, ParEGO-KG and PALS.

a good performance level on both metrics, allowing for better Pareto set and front estimations in most cases.

A summary of the numerical results obtained when comparing the proposed PALS approach and the reference methods over the set of benchmark problems is presented in Table 1. PALS consistently outperforms the other methods in terms of the ability to better estimate the Pareto set (as seen through the misclassification rate metric). For Pareto front estimation (see volume of the symmetric difference metric), the most efficient algorithm depends on the problem considered, with PALS presenting an overall very good performance—although not being the best performer for problems g_7 and g_9, it places second with a metric value close to the leader. Additionally, PALS performs significantly better than pure random search on both metrics for all problems.

Table 1: Average metrics (value in percentage), for the volume of the symmetric difference (V_d) and misclassification rate (M) at final iteration, for PRS, CoRS, ParEGO-EI$_m$, ParEGO-KG, and PALS. The best metric values are highlighted in bold with a gray background. Metrics at 10% of the best metric are highlighted with a light gray background and considered as acceptable performance.

	PRS		CoRS		ParEGO-EI$_m$		ParEGO-KG		PALS	
g	V_d	M	V_d	M	V_d	M	V_d	M	V_d	M
g_1	0.793	7.568	0.660	7.501	0.786	11.418	0.687	6.211	**0.596**	**6.061**
g_2	0.968	1.222	0.712	1.058	0.638	1.330	0.502	1.103	**0.443**	**0.966**
g_3	1.098	3.491	1.143	3.456	0.707	3.388	0.711	3.019	**0.700**	**2.930**
g_4	1.245	2.050	1.214	2.180	1.025	2.203	0.756	1.827	**0.744**	**1.594**
g_5	1.076	3.815	0.749	3.321	0.920	7.975	0.683	5.743	**0.594**	**2.842**
g_6	1.451	0.712	0.727	0.391	0.467	1.545	0.429	1.049	**0.394**	**0.383**
g_7	0.872	2.492	0.624	2.539	0.512	4.675	**0.295**	3.022	0.408	**2.230**
g_8	0.868	4.553	0.776	4.752	0.731	6.517	0.568	4.320	**0.552**	**3.658**
g_9	1.068	1.471	0.694	1.169	0.639	2.720	**0.359**	1.466	0.385	**0.850**

6. Conclusions and future work

In this article, an extension of the Pareto Active Learning (PAL) algorithm was presented, which makes it suitable for stochastic simulators with possibly high output variability. This new version was named Pareto Active Learning for Stochastic Simulators (PALS) and showed promising results in a numerical benchmark on nine bi-objective test problems in dimension two. Performance was compared with a Pure Random Search (PRS) approach, a Concentrated Random Sampling (CoRS) approach and two modified ParEGO algorithms. For the considered benchmark, PALS is the best algorithm to estimate the Pareto set solutions of the problem. It is also the best performer for Pareto front estimation for most of the problems. The modified ParEGO algorithm named ParEGO-KG also presents promising results when considering the Pareto front estimation.

We believe that, for future work, comparing the proposed algorithm with alternative approaches such as SK-MOCBA (Rojas Gonzalez et al., 2020) or PESMO (Hernández-Lobato et al., 2016) should be envisaged. Additionally, we could be interested in comparisons with algorithms from the Multi-objective Evolutionary Algorithms (MOEA) literature. Interesting perspectives could also include creating hybrid algorithms that could leverage the different advantages of each algorithm.

A difference in PALS with regards to PAL is that each point selected during the optimization process is evaluated several times. This is especially advantageous when multiple parallel computing units are available or when the infill criterion to choose the next point to evaluate is expensive to compute with regards to the actual simulation time. Although the numerical experiments have shown no significant difference at final iteration between different batch sizes, the intermediate performances were different. Results show that smaller batch sizes have better performance at initial stages, which seems to indicate a possible

interest in having a noise-level dependent adaptive batch size (Lyu & Ludkovski, 2022) during the optimization process.

A main component in the PAL approach is the β parameter that defines the size of the uncertainty region considered around the model's prediction. In the original work, an increasing β is proposed. Large parameter values tend to promote exploration, while smaller values promote exploitation. Numerical experiments have shown, for the problems considered, that selecting a constant value for the parameter could provide better results than the original approach. Nevertheless, the concept of using an increasing β parameter seems relevant, but further studies should be performed to determine a suitable initial value and rate of change depending on the problem structure. Therefore, the ideal β selection seems to depend on the problem and noise level, and should still be further studied.

Finally, even though PALS is an algorithm suitable for the case where the input space is finite, we believe that the main idea could be applied—with some adaptations—to continuous input spaces.

References

Astudillo, R., & Frazier, P. I. (2017). Multi-attribute Bayesian optimization under utility uncertainty. In *NIPS Workshop on Bayesian Optimization, BayesOpt 2017*. Long Beach, CA, USA. 5 pages.

Belakaria, S., Deshwal, A., & Doppa, J. R. (2020). Max-value entropy search for multi-objective Bayesian optimization with constraints. In *Machine Learning and the Physical Sciences, NeurIPS ML4PS 2020*. 8 pages.

Binois, M., Ginsbourger, D., & Roustant, O. (2015). Quantifying uncertainty on Pareto fronts with Gaussian process conditional simulations. *European Journal of Operational Research*, *243*, 386–394.

Binois, M., Gramacy, R. B., & Ludkovski, M. (2018). Practical heteroscedastic Gaussian process modeling for large simulation experiments. *Journal of Computational and Graphical Statistics*, *27*, 808–821.

Binois, M., Huang, J., Gramacy, R. B., & Ludkovski, M. (2019). Replication or exploration: Sequential design for stochastic simulation experiments. *Technometrics*, *61*, 7–23.

Branke, J., & Zhang, W. (2015). A new myopic sequential sampling algorithm for multi-objective problems. In L. Yilmaz, W. K. V. Chan, I. Moon, T. M. K. Roeder, C. Macal, & M. D. Rossetti (Eds.), *Proceedings of the 2015 Winter Simulation Conference. December 6–9, Huntington Beach, CA, USA.*. IEEE.

Chiles, J.-P., & Delfiner, P. (1999). *Geostatistics: modeling spatial uncertainty*. John Wiley & Sons.

Dutrieux, H., Aleksovska, I., Bect, J., Vazquez, E., Delille, G., & François, B. (2015a). The informational approach to global optimization in presence of very noisy evaluation results. Application to the optimization of renewable energy integration strategies. In *47èmes Journées de Statistique de la SFdS*.

Dutrieux, H., Delille, G., François, B., & Malarange, G. (2015b). An innovative method to assess solutions for integrating renewable generation into distribution networks over multi-year horizons. In *23rd International Conference and Exhibition on Electricity Distribution (CIRED 2015). June 15–18, Lyon, France*.

Emmerich, M. T. M., Giannakoglou, K. C., & Naujoks, B. (2006). Single-and multiobjective evolutionary optimization assisted by Gaussian random field metamodels. *IEEE Transactions on Evolutionary Computation*, *10*, 421–439.

Feliot, P., Bect, J., & Vazquez, E. (2017). A Bayesian approach to constrained single- and multi-objective optimization. *Journal of Global Optimization*, *67*, 97–133.

Frazier, P., Powell, W., & Dayanik, S. (2009). The knowledge-gradient policy for correlated normal beliefs. *INFORMS Journal on Computing*, *21*, 599–613.

Habib, A., Singh, H. K., & Ray, T. (2016). A multi-objective batch infill strategy for efficient global optimization. In *2016 IEEE Congress on Evolutionary Computation. July 24–29, Vancouver, Canada.* (pp. 4336–4343). IEEE.

Hernández-Lobato, D., Hernández-Lobato, J. M., Shah, A., & Adams, R. P. (2016). Predictive Entropy Search for Multi-objective Bayesian Optimization with Constraints. In *33rd International Conference on Machine Learning. June 20–22, New York, New York, USA* (pp. 1492–1501). PMLR volume 48 of *Proceedings of Machine Learning Research*.

Hunter, S. R., Applegate, E. A., Arora, V., Chong, B., Cooper, K., Rincón-Guevara, O., & Vivas-Valencia, C. (2019). An introduction to multiobjective simulation optimization. *ACM Transactions on Modeling and Computer Simulation*, *29*, 1–36.

Hunter, S. R., & McClosky, B. (2016). Maximizing quantitative traits in the mating design problem via simulation-based Pareto estimation. *IIE Transactions*, *48*, 565–578.

Jones, D. R., Schonlau, M., & Welch, W. J. (1998). Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, *13*, 455–492.

Knowles, J. (2006). ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Transactions on Evolutionary Computation*, *10*, 50–66.

Lee, L. H., Chew, E. P., Teng, S., & Goldsman, D. (2010). Finding the non-dominated Pareto set for multi-objective simulation models. *IIE Transactions*, *42*, 656–674.

Lyu, X., & Ludkovski, M. (2022). Adaptive batching for Gaussian process surrogates with application in noisy level set estimation. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, *15*, 225–246.

Matheron, G. (1963). Principles of geostatistics. *Economic Geology*, *58*, 1246–1266.

Mattila, V., & Virtanen, K. (2014). Maintenance scheduling of a fleet of fighter aircraft through multi-objective simulation-optimization. *Simulation*, *90*, 1023–1040.

Pasupathy, R., Hunter, S. R., Pujowidianto, N. A., Lee, L. H., & Chen, C. H. (2014). Stochastically constrained ranking and selection via score. *ACM Transactions on Modeling and Computer Simulation*, *25*, 1–26.

Picheny, V. (2015). Multiobjective optimization using Gaussian process emulators via stepwise uncertainty reduction. *Statistics and Computing*, *25*, 1265–1280.

Rasmussen, C. E., & Williams, C. K. I. (2006). *Gaussian Processes for Machine Learning*. MIT Press.

Rojas Gonzalez, S., Jalali, H., & Van Nieuwenhuyse, I. (2020). A multiobjective stochastic simulation optimization algorithm. *European Journal of Operational Research*, *284*, 212–226.

Santner, T. J., Notz, W. I., & Williams, B. J. (2003). *The Design and Analysis of Computer Experiments*. Springer.

Spearmint (2016). Spearmint: Bayesian optimization codebase - PESM branch. https://github.com/HIPS/Spearmint/tree/PESM (accessed: 2021-03-30).

Srinivas, N., Krause, A., Kakade, S., & Seeger, M. (2010). Gaussian process optimization in the bandit setting: no regret and experimental design. In J. Fürnkranz, & T. Joachims (Eds.), *Proceedings of the 27th International Conference on Machine Learning. June 21–24, Haifa, Israel* (pp. 1015–1022). Omnipress.

Vazquez, E., Villemonteix, J., Sidorkiewicz, M., & Walter, É. (2008). Global optimization based on noisy evaluations: An empirical study of two statistical approaches. *Journal of Physics: Conference Series*, *135*.

Zuluaga, M., Krause, A., Sergent, G., & Püschel, M. (2013). Active learning for multi-objective optimization. In S. Dasgupta, & D. McAllester (Eds.), *Proceedings of the 30th International Conference on Machine Learning. June 17–19, Atlanta, Georgia, USA* (pp. 462–470). PMLR volume 28 of *Proceedings of Machine Learning Research*.

Table A.1: Definition of functions f_1 to f_{15}

f	Definition
f_1	$780000 + 110000x_1 - 12000x_2 - 36000x_1x_2 + 280000x_1^2 + 50000x_2^2$
f_2	$0.83 + 0.17x_1 - 0.015x_2 - 0.0038x_1x_2 + 0.061x_1^2 + 0.0011x_2^2$
f_3	$e^{(0.36(x_1+x_2))} + 0.6x_1 + 1.2x_2^2 + 3\sin(0.8\pi x_1)$
f_4	$(x_2 - a * x_1 .* x_1 + b * x_1 - 6).^2 + c * \cos(x_1) + 10$
f_5	$100(x_2 - x_1^2)^2 + (1 - x_1)^2$
f_6 to f_{15}	$c_1 + c_2x_1 + c_3x_2 + c_4x_1x_2$ $+ c_5x_1^2 + c_6x_2^2 + c_7x_1^2x_2 + c_8x_1x_2^2 + c_9x_1^3 + c_{10}x_2^3$

Table A.2: Coefficients used for functions f_6 to f_{15}

	Function coefficients									
	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}
c_1	0.36	0.68	0.094	0.61	-0.38	-0.19	0.78	-0.45	-0.45	0.75
c_2	8.1	-9.4	-7.2	5	8.5	4.8	6	7.8	-9.3	7.4
c_3	7.5	9.1	7	2.3	1.4	2.1	-4.7	-7.7	-3.5	-8.2
c_4	-83	-2.9	49	-5.3	63	42	90	28	14	-98
c_5	26	-60	68	30	81	56	-85	34	-9.7	15
c_6	-80	72	-49	-66	96	77	-82	-31	22	-31
c_7	-440	160	630	-170	-120	410	600	-500	-880	-450
c_8	94	-830	-510	-99	-780	360	890	-170	-370	-62
c_9	920	-580	860	-830	-480	150	370	-480	550	780
c_{10}	930	-920	-300	430	-180	-16	-740	530	390	-260

Appendix A. Benchmark problems

This section introduces the test problems used for the numerical experiments of Section 5.

All the test functions are presented in Table A.1 and are defined in $[0,1]^2$. Functions f_1 and f_2 mimick a real-life problem presented by Dutrieux et al. (2015b). Functions f_3, f_4 (Branin) and f_5 (Rosenbrock) are well-known test functions, rescaled to $[0,1]^2$. Functions f_6 to f_{15} correspond to randomly generated third degree polynomials, whose coefficients are presented in Table A.2. In our experiments, the values of these 15 test functions are scaled to $[0,1]$, assuming that the minimum and maximum values are known (cf. Remark 2).

A total of nine problems denoted by g_1 to g_9 are proposed based on the previously defined functions (see Table A.3). For each problem, the input set $\mathbb{X}$ is a 21×21 regular grid on $[0,1]^2$. The Pareto fronts are presented in Figure A.1 and the Pareto sets in Figure A.2.

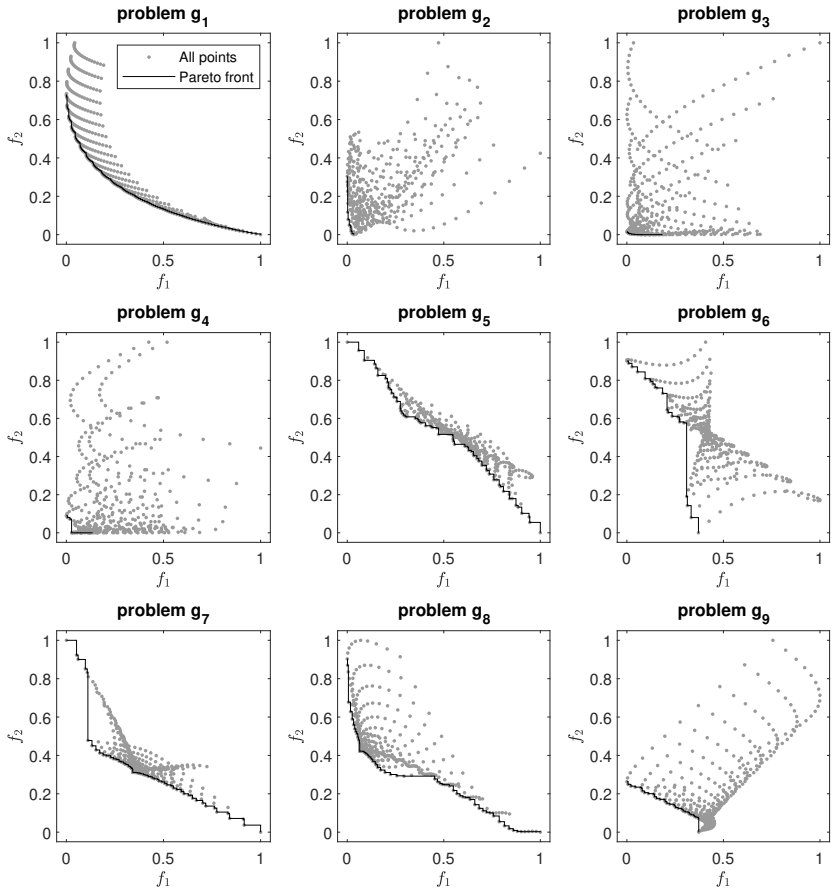

Figure A.1: Pareto front solutions for test problems.

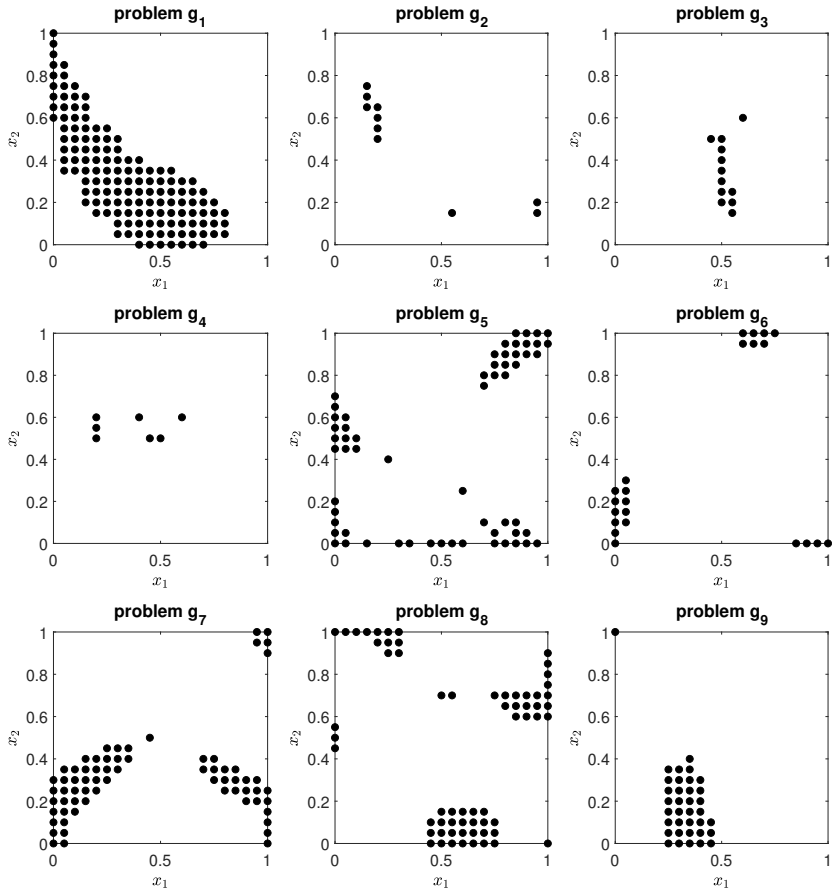

Figure A.2: Pareto set solutions for test problems.

Table A.3: Definition of nine test problems using the test functions presented in Tables A.1–A.2. For problems g_5 to g_9 the input argument is shifted by x_0: if function f_k is indicated in the table, then the actual objective function is $x \mapsto f_k(x - x_0)$. The last two columns provide respectively the cardinality of the Pareto set and the proportion of Pareto-optimal solutions in the 21x21 input set.

	objectives		noise variance				
g	f_1^g	f_2^g	σ_1^2	σ_2^2	x_0	$\mid \mathcal{P} \mid$	$\mid \mathcal{P} \mid / \mid \mathbb{X} \mid$
g_1	f_1	f_2	3.6×10^9	3.9×10^{-3}		136	31%
g_2	f_4	f_3	3.1×10^2	4.8×10^3		10	2%
g_3	f_4	f_5	3.1×10^2	5.7×10^8		12	3%
g_4	f_3	f_5	4.8×10^3	5.7×10^8		7	2%
g_5	f_6	f_7	7.0×10^2	5.6×10^3	$(0.5, 0.5)$	60	14%
g_6	f_8	f_9	5.8×10^2	3.1×10^3	$(0.5, 0.5)$	22	5%
g_7	f_{10}	f_{11}	2.1×10^3	3.2×10^2	$(0.5, 0.5)$	67	15%
g_8	f_{12}	f_{13}	1.4×10^4	1.6×10^3	$(0.3, 0.8)(0.6, 0.6)$	63	14%
g_9	f_{14}	f_{15}	3.7×10^3	2.0×10^4	$(0.3, 0.8)$	36	8%

Appendix B. Performance metrics

We introduce the performance metrics proposed to assess prediction quality. The volume of the symmetric difference measures a global error in the estimation. We use it to assess Pareto front estimates. The misclassification rate is used as a global error metric to evaluate Pareto set estimates.

Appendix B.1. Volume of the symmetric difference

Let the "dominated region" defined between a Pareto front $\mathcal{F}$ and a reference point $\boldsymbol{R}$ be the set of all the points that simultaneously dominate $\boldsymbol{R}$, and are dominated by a member of $\mathcal{F}$: $D(\mathcal{F}) = \cup_{y^\star \in \mathcal{F}} \{\boldsymbol{y} \in \mathbb{R}^q : \boldsymbol{y}^\star \prec \boldsymbol{y} \prec \boldsymbol{R}\}$. Recalling that our objective functions are all scaled to $[0, 1]$, we take $\boldsymbol{R} = (1.1, 1.1)$ in our (bi-objective) experiments. The volume of the dominated region is denoted by $V(\mathcal{F})$, and is illustrated in Figure B.1 (left).

We can then define the volume of the symmetric difference of the hypervolumes defined between the true Pareto front and the estimation, and a reference point $\boldsymbol{R}$. We denote it $V_d(\mathcal{F}^\star, \hat{\mathcal{F}})$:

$$V_d(\mathcal{F}^\star, \hat{\mathcal{F}}) = V(D(\mathcal{F}) \setminus D(\mathcal{F}^\star)) \cup (D(\mathcal{F}^\star) \setminus D(\mathcal{F})), \tag{B.1}$$

and illustrate it in Figure B.1 (right):

Appendix B.2. Misclassification rate

The misclassification rate provides a measure of the error of a Pareto Set prediction. Consider a true Pareto set $\mathcal{P}^\star$ and a Pareto set prediction $\widehat{\mathcal{P}}$. Then

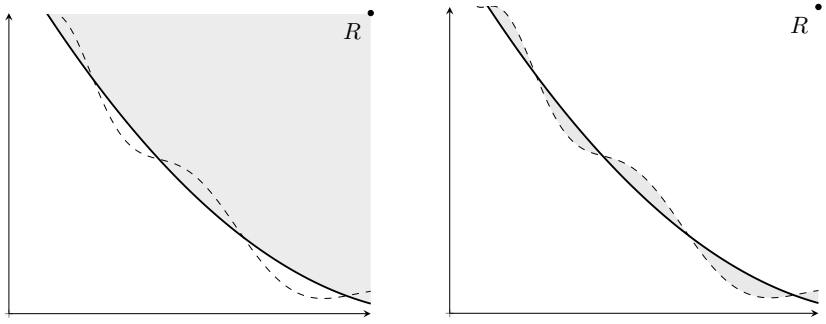

Figure B.1: Hypervolume generated (gray area) between a Pareto front prediction $\hat{\mathcal{F}}$ (filled line) and a reference point R, in a bi-objective example (left), and volume of the symmetric difference (gray area) between a Pareto front $\mathcal{F}$ (dashed line) and a prediction $\hat{\mathcal{F}}$ (filled line) using reference point R, in a bi-objective example (right).

the misclassification rate $M(\mathcal{P}^\star, \widehat{\mathcal{P}})$ is defined as the proportion of points that differ between the prediction and true sets:

$$M(\mathcal{P}^\star, \widehat{\mathcal{P}}) = \frac{1}{|\mathbb{X}|} \sum_{x \in \mathbb{X}} \left(\mathbb{1}_{x \in \mathcal{P}^\star} - \mathbb{1}_{x \in \widehat{\mathcal{P}}} \right)^2 . \tag{B.2}$$