

Chapter 5

Exotic Bayesian Optimization



5.1 Constrained Global Optimization

GO and BO have been considered first in “essentially unconstrained” conditions, where the solution was searched for within a bounded-box search space. Recently, due to methodological and application reasons, there has been an increasing interest in constrained global optimization (CGO).

$$x^* = \arg \min_{x \in X} f(x)$$

Subject to

$$c_i(x) \leq 0 \quad i = 1, \dots, n_c$$

A paper of general interest about a taxonomy of constraints in simulation-based optimization is given in Le Digabel (2015). Some general remarks have to be taken into account:

1. Even if the constraints are analytically defined, they presence without the convexity assumption raises a whole new set of challenging topics, in particular, the interaction between the feasible region and the surrogate probabilistic model.
2. Since BO assumes the objective function as black box, a natural extension is to consider the constraints as black box as well (unknown constraints).
3. A relevant case is when the objective function is undefined, and cannot be therefore computed, outside the feasible region. In this case, we speak about partially defined objective functions (Rudenko 1994; Sergeyev et al. 2007).

Point three is particularly relevant when the evaluation of the objective requires, as in black box conditions, the execution of a simulation model which can return a valid output or a failure message. An example of failure can be a computational fluid dynamics solver that does not converge due to instability of the numerical scheme (Sacher et al. 2018) or a hydraulic simulator if the input generates pressures or flows physically impossible (Tsai et al. 2018). Also frequent is the case of a

complex machine learning model where the training of the model, and therefore the evaluation of the loss function, relies on gradient (stochastic) descent which may fail to converge. Such an event prevents the exploration of a neighbourhood of a “not computable” point and halts the sequential optimization procedure. A naïve solution, in presence of a not computable point, is to associate to it a fixed high (low) penalty value for the objective function to be minimized (maximized): still, determining suitable value is not a trivial task and might imply, anyway, a loss of accuracy in the Gaussian surrogate model and of sample efficiency.

There have been several attempts at leveraging BO framework into dealing with constrained optimization: the main problem is to propose an acquisition function for CBO.

The use of GP-and EI-based heuristics has been first proposed in Jones et al. (1998) allowing for $c_i(x)$ to be black box and assuming their mutual independence and with the objective function. A GP is given as a prior to each constraint. If f_c^+ is the best feasible observation of f , the EI acquisition function is:

$$\text{EI}(x|f_c^+) = \begin{cases} (\mu(x) - f_c^+) \Phi(Z) + \sigma(x)\phi(Z) & \text{if } \sigma(x) > 0 \\ 0 & \text{if } \sigma(x) = 0 \end{cases}$$

where ϕ and Φ are the probability distribution and the cumulative distribution functions, respectively, and

$$Z = \begin{cases} \frac{\mu(x) - f_c^+}{\sigma(x)} & \text{if } \sigma(x) > 0 \\ 0 & \text{if } \sigma(x) = 0 \end{cases}$$

In presence of constraints the formula becomes:

$$\text{EIC}(x|f_c^+) = \text{EI}(x|f_c^+) \prod_{i=1}^{n_c} \mathbb{P}(c_i(x) \leq 0)$$

where the improvement of a candidate solution x over f is zero if x is not feasible. When the noise in the constraints is taken into account, we may not know which observations are feasible: for instance, the best GP mean value satisfying each constraint $c_i(x)$ with a probability at least $1 - \delta_i$ can be used (Letham et al. 2019). Gramacy (2016) proposes a different approach for handling constraints in which they are brought into the objective function via a Lagrangian. EI is no longer tractable analytically but can be evaluated numerically via Monte Carlo integration or quadrature (Picheny 2016). Relevant prior results on BO with unknown constraints proposed new acquisition functions, such as integrated expected conditioned improvement (IECI) (Gramacy and Lee 2011). Another approach is presented in Feliot et al. (2017) where an adaptive Random Search is used to approximate feasible region while optimizing the objective function. A penalty approach has been considered first in Gardner et al. (2014), where a penalty is assigned directly to the acquisition function in case of infeasibility, with the aim to move away from infeasible regions. A similar approach

has been extended in Candelieri et al. (2018) also in the case in which the function is partially defined: infeasibility is treated by assigning a fixed penalty as value of the objective function (we refer as “BO with penalty”).

A new general approach is offered by information-based methods, which have been extended to the constrained case (e.g. predictive entropy search with constraints, PESC) in Hernandez-Lobato et al. (2015): the code for PESC is included in Spearmint and available at <https://github.com/HIPS/Spearmint/tree/PESC>.

The above approaches assume that the number of constraints is known a priori and they are statistically independent.

The assumption of independence permits to compute the probability of feasibility simply as the product of individual probabilities with respect to every constraint. The result is multiplied by the acquisition function whose optimization would prefer points satisfying the constraints with high probability.

The issue of partially defined function or equivalently “crash constraints” or “non-computable domains” transforms the GP-based feasibility evaluation from a regression problem into a classification one. This classification approach has been analyzed in several papers and discussed in the following.

In Basudhar et al. (2012) a probabilistic SVM (PSVM) is used to calculate the so-called probability of feasibility and the optimization scheme alternates between a global search for the optimal solution, depending on both this probability and the estimated value of the objective function—modelled through a GP—and a local refinement of the PSVM through an adaptive local sampling scheme. The most common PSVM model is based on the sigmoid function (Vapnik 1998). For a given sample x , the probability of belonging to the +1 class (i.e. “feasible”) is:

$$P(+1|x) = \frac{1}{1 + e^{As(x)+B}}$$

The parameters A ($A < 0$) and B of the sigmoid function are found by maximum likelihood. Two alternative formulations of the acquisition function are proposed:

$$\max_x EI(x)P(+1|x)$$

and

$$\begin{aligned} & \max_x EI(x) \\ & \text{Subject to} \\ & P(+1|x) \geq 0.5 \end{aligned}$$

Other approaches, also based on GP, and focused on cases where the objective function cannot be computed (not computable domains or crash constraints) have been proposed in Sacher et al. (2018); Bachoc et al. (2019).

The previous approaches use independent GPs to model the objective function and the constraints, requiring two strong assumptions: a priori knowledge about the

number of constraints and the independence among objective function and all the constraints. Extending EI to account for correlations between constraints and the objective function is still, according to Letham et al. (2019), an open challenge.

To overcome this limitation, a new approach, namely SVM-CBO (Support Vector Machine based Constrained BO), has been proposed in Candelieri and Archetti (2019): the main contribution of the paper is the development of a method which does not require any of the two previous assumptions. The approach uses support vector machine (SVM) to sequentially estimate and model the unknown feasible region (Ω) within the search space (i.e. feasibility determination), without any assumption on the number of constraints as well as their independence.

SVM-CBO is organized in two phases: the first is aimed to provide a first estimate of Ω (feasibility determination) and the second is BO performed on such an estimate, only. The motivation is that we are interested in obtaining a good approximation of the overall feasible region and not only close to the optimal solution (i.e. feasibility determination is a goal per se, in our approach). Another relevant difference with Basudhar et al. (2012) is that SVM-CBO uses more efficiently the available “budget” (i.e. maximum number of function evaluations): at every iteration, of both phase 1 and 2, we perform just one function evaluation, while in the boundary refinement of Basudhar et al. (2012) a given number n_p of function evaluations is performed in the neighbourhood of the next point to evaluate, with the aim to locally refine the boundary estimate. SVM-CBO is detailed in the following section.

5.2 Support Vector Machine—Constrained Bayesian Optimization

We start with the definition of the problem, that is:

$$\min_{x \in \Omega \subset X \subset \mathbb{R}^d} f(x)$$

where $f(x)$ has the following properties: it is black box, multi-extremal, expensive and partially defined. The last feature means that $f(x)$ is undefined outside that feasible region Ω , which is a subset of overall bounded-box search space $X \subset \mathbb{R}^d$. Moreover, we consider the case that constraints defining the feasible region are also black box.

We introduce some notation that will be used in the following:

- $D_n^\Omega = \{(x_i, y_i)\}_{i=1, \dots, n}$ is the **feasibility determination** dataset;
- $D_l^f = \{(x_i, f(x_i))\}_{i=1, \dots, l}$ is the **function evaluations** dataset, with $l \leq n$ (because $f(x)$ is partially defined on X) and where l is the number of points where it was possible to compute f out of the n queried so far;

and x_i is the i -th queried point and $y_i = \{+1, -1\}$ defines if x_i is feasible or infeasible, respectively.

Phase 1—Feasibility determination

The first phase of the approach aims to find an estimate $\tilde{\Omega}$ of the actual feasible region Ω in M function evaluations ($\tilde{\Omega}_M = \tilde{\Omega}$). The sequence of function evaluations is determined according to an SMBO process where the surrogate model—of the feasible region, in this phase—provides the currently estimated feasible region $\tilde{\Omega}_n$. As surrogate model we use the (non-linear) separation hyperplane of an SVM classifier, trained on the set D_n^Ω . The SVM classifier uses an RBF kernel to model feasible regions with non-linear boundaries.

Let denote with $h_n(x)$ the argument of the SVM-based classification function:

$$h_n(x) = \sum_{i=1}^{n_{SV}} \alpha_i y_i k(\bar{x}_i, x) + b$$

where α_i and y_i are the Lagrangian coefficient and the “feasibility label” of the i -th support vector, \bar{x}_i , respectively, $k(\cdot, \cdot)$ is the kernel function (i.e. an RBF kernel, in this study), b is the offset and n_{SV} is the number of support vectors.

The boundaries of the estimated feasible region $\tilde{\Omega}_n$ are given by $h_n(x) = 0$ (i.e. non-linear separation hyperplane). The SVM-based classification function provides the estimated feasibility for any $x \in X$:

$$\tilde{y} = \text{sign}(h_n(x)) = \begin{cases} +1 & \text{if } x \in \tilde{\Omega}_n \\ -1 & \text{if } x \notin \tilde{\Omega}_n \end{cases}$$

With respect to the aim of the first phase, we propose a “*feasibility acquisition function*” aimed at identifying the next promising point according to two different goals:

- Improving the estimate of feasible region
- Discovering possible disconnected feasible regions.

To deal with the first goal, we use the distance of x from the boundaries of the currently estimated feasible region $\tilde{\Omega}_n$, using the following formula from the SVM classification theory:

$$d_n(h_n(x), x) = |h_n(x)| = \left| \sum_{i=1}^{n_{SV}} \alpha_i y_i k(\bar{x}_i, x) + b \right|$$

To deal with the second goal, we introduce the concept of “coverage of the search space”, defined by:

$$c_n(x) = \sum_{i=1}^n e^{-\frac{\|\bar{x}_i - x\|^2}{2\sigma_c^2}}$$

So, $c_n(x)$ is a sum of n RBF functions centred on the points evaluated so far, with σ_c a parameter to set the width of the corresponding bell-shaped curve.

Finally, the *feasibility acquisition function* is given by the sum of $d_n(h_n(x), x)$ and $c_n(x)$, and the next promising point is identified by solving the following optimization problem:

$$x_{n+1} = \underset{x \in X}{\operatorname{argmin}} \{d_n(h_n(x), x) + c_n(x)\}$$

Thus, we want to select the point associated with the minimal distance from the boundaries of the current estimated feasible region and the minimal coverage (i.e. max uncertainty). This allows us to balance between improving the estimate of the feasible region and discovering possible disconnected feasible regions (in less explored areas of the search space). It is important to highlight that, in phase 1, the optimization is performed on the overall bounded-box search space X .

After the function evaluation of the new point x_{n+1} , the following information is available:

$$y_{n+1} = \begin{cases} +1 & \text{if } x_{n+1} \in \Omega; f(x_{n+1}) \text{ is defined} \\ -1 & \text{if } x_{n+1} \notin \Omega : f(x_{n+1}) \text{ is not defined} \end{cases}$$

and the following updates are performed:

- Feasibility determination dataset and estimated feasible region $\tilde{\Omega}_{n+1}$

$$\begin{aligned} D_{n+1}^{\Omega} &= D_n^{\Omega} \cup \{(x_{n+1}, y_{n+1})\} \\ &h_{n+1}(x) | D_{n+1}^{\Omega} \\ n &\leftarrow n + 1 \end{aligned}$$

- Only if $x \in \Omega$, function evaluations dataset

$$\begin{aligned} D_{l+1}^f &= D_l^f \cup \{(x_{l+1}, f(x_{l+1}))\} \\ l &\leftarrow l + 1 \end{aligned}$$

The SMBO process for phase 1 is repeated until $n = M$.

Phase 2—Bayesian Optimization in the estimated feasible region

In this phase, a traditional BO process is performed but with the following relevant differences:

- the search space is not box-bounded but the estimated feasible region $\tilde{\Omega}_n$ identified in phase 1

- the surrogate model—a GP—is fitted only using the feasible solutions observed so far, D_l^f
- the acquisition function for phase 2—lower confidence bound (LCB), in this study—is defined on $\tilde{\Omega}_n$, only

Thus, the next point to evaluate is given by:

$$x_{n+1} = \operatorname{argmin}_{x \in \tilde{\Omega}_n} \{\operatorname{LCB}_n(x) = \mu_n(x) - \beta_n \sigma_n(x)\}$$

where $\mu_n(x)$ and $\sigma_n(x)$ are the mean and the standard deviation of the current GP-based surrogate model and β_n is the inflate parameter to deal with the trade-off between exploration and exploitation for this phase. It is important to highlight that, contrary to phase 1, the acquisition function is here minimized on $\tilde{\Omega}_n$, only, instead of the entire bounded-box search domain X .

The point x_{n+1} is just expected to be feasible, according to $\tilde{\Omega}_n$ but the information on its actual feasibility is known only after having checked whether $f(x_{n+1})$ can or cannot be computed (i.e. it is defined or not in x_{n+1}). Subsequently, the feasibility determination dataset is updated as follows:

$$D_{n+1}^\Omega = D_n^{\Omega} \cup \{(x_{n+1}, y_{n+1})\}$$

and according to the two alternative cases:

- x_{n+1} is actually feasible: $x_{n+1} \in \Omega$, $y_{n+1} = +1$;

the function evaluations dataset is updated as follows: $D_{l+1}^f = D_l^f \cup \{(x_{l+1}, f(x_{l+1}))\}$, with $l \leq n$ is the number of the feasible solutions with respect to all the points observed. The current estimated feasible region $\tilde{\Omega}_n$ can be considered accurate and retraining of the SVM classifier can be avoided: $\tilde{\Omega}_{n+1} = \tilde{\Omega}_n$

- x_{n+1} is actually infeasible: $x_{n+1} \notin \Omega$, $y_{n+1} = -1$;

the estimated feasible region must be updated to reduce the risk for further infeasible evaluations

$$h_{n+1}(x) | D_{l+1}^f \Rightarrow \tilde{\Omega}_{n+1}$$

The phase 2 continues until the overall available budget $n = N$ is reached.

In Candelieri (2019) the SVM-CBO approach has been validated on five 2D test functions for CGO and compared to “BO with penalty”. An overall budget of 100 function evaluations, divided as follows:

- 10 for initialization through Latin hypercube sampling (LHS);
- 60 for feasibility estimation (phase 1)
- and 30 for SMBO constrained to estimated feasible region (phase 2).

In the case of BO with penalty, the same budget has been divided as follows:

- 10 evaluations for initialization (LHS)
- and 90 for BO (that is the sum of budget used for phase 1 and phase 2 in the proposed approach).

For each independent run, the initial set of solutions identified through LHS is the same for SVM-CBO and BO with penalty, in order to avoid differences in the values of the gap metric due to different initialization.

The so-called *Gap metric* has been used to measure the improvement obtained along the SMBO process with respect to global optimum $f(x^*)$ and the initial best solution $f(x_0)$ obtained from the initialization step:

$$G_n = \frac{|f(x_0) - f(x^+)|}{|f(x_0) - f(x^*)|}$$

where $f(x^+)$ is the “best seen” up to iteration n . Gap metrics varies in the range $[0, 1]$. For statistical significance, the gap metrics have been computed on 30 different runs, performed for every test function and for both SVM-CBO and BO with penalty.

While gap metric allows for comparing SVM-CBO to BO with penalty, it was important to introduce another performance measure to quantify how good is the approximation of the feasible region, along the SMBO process. We have defined a simple *overlap metric* as follows:

$$O_n(\Omega, \tilde{\Omega}_n) = \begin{cases} \frac{\text{Vol}(\tilde{\Omega}_n \cap \Omega)}{\text{Vol}(\Omega)} & \text{if } \text{Vol}(\tilde{\Omega}_n) < \text{Vol}(\Omega) \vee (\text{Vol}(\tilde{\Omega}_n \cap \Omega) \\ & = \text{Vol}(\Omega) \wedge \text{Vol}(\tilde{\Omega}_n) = \text{Vol}(\Omega)) \\ \frac{\text{Vol}(\tilde{\Omega}_n)}{\text{Vol}(\Omega)} & \text{otherwise} \end{cases}$$

where Ω and $\tilde{\Omega}_n$ are the actual and the estimated feasible regions, respectively, and $\text{Vol}()$ is the volume of a region. More precisely, the volume of a region is computed as an approximation by simply generating a grid of points within the search space and then counting the number of points falling into that region.

According to its definition, the overlap metric can vary in the range $[0, \infty)$, where a good approximation of the feasible region is associated with a value equal to 1 while no overlap is associated with a value equal to 0. Values higher than 1 represent situations where $\Omega \subset \tilde{\Omega}_n$.

The following set of figures show the gap metric computed for every test function with respect to the number of function evaluations, excluding the initialization through LHS. The value of gap metric at iteration 0 is the best seen at the end of the initialization step (i.e. $f(x_0)$ in the gap metric formula). Each graph compares the gap metric provided by SVM-CBO and BO with penalty, respectively. Both the average and the standard deviation of the gap metric, computed on 30 independent runs for each approach, are depicted. The higher effectiveness of the proposed approach is clear, even considering the variance in the performances. The end of phase 1 is represented by a vertical dotted line in the charts; a significant improvement of the

SVM-CBO’s gap metric is observed after this phase, in every test case. It is important to remind that phase 1 of the SVM-CBO is aimed at approximating the unknown feasible region, while the optimization process only starts with phase 2. Thus, the relevant shift in the gap metric is motivated by the explicit model of the feasible region learned in phase 1 (Fig. 5.1).

Finally, the following set of figures show how the *overlap metric* changes sequentially over the SMBO process. Since 10 initial function evaluations are all used to provide a first estimate of the feasible region, the x -axis is limited to 90, that is the portion of the budget allocated for phase 1 and phase 2: during these phases—particularly during phase 1—the estimate of the feasible region is updated and the overlap metric changes consequently. The dotted line represents the mean overlap

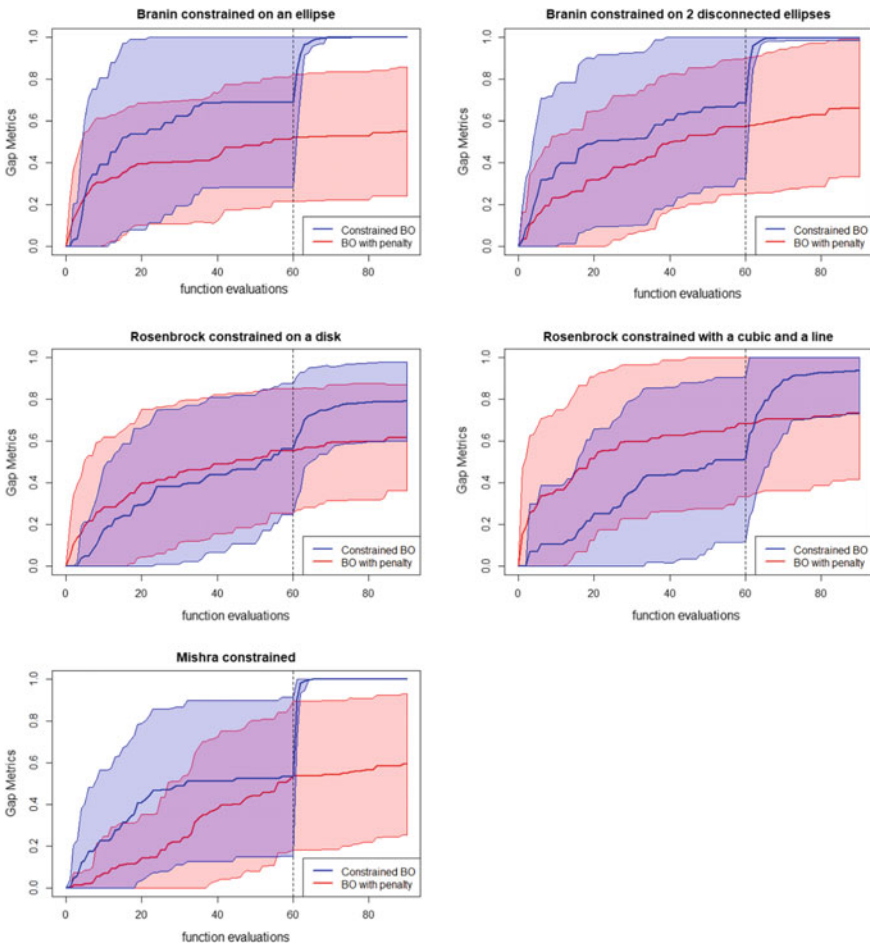


Fig. 5.1 Comparison of gap metrics for the proposed SVM-CBO approach versus BO with penalty, for each one of the five test functions, respectively

metric and the shaded area represents the standard deviation. The figures show that the proposed SVM-CBO approach achieves a reasonable overlap (higher than 77%) between estimated and actual feasible region, already after 30–35 function evaluations. The most difficult case was the one with a feasible region consisting of two disconnected ellipses, but, before the end of budget, overlap metric achieved anyway the value of 80% (Fig. 5.2).

SVM-CBO offers a good estimation of the feasible search space, even for complex feasible search spaces consisting of disconnected regions. Seldom, the BO process, in phase 2, suggested infeasible points but, thanks to the update of the estimated feasible region, it was able to quickly fall close to the optimum, without wasting

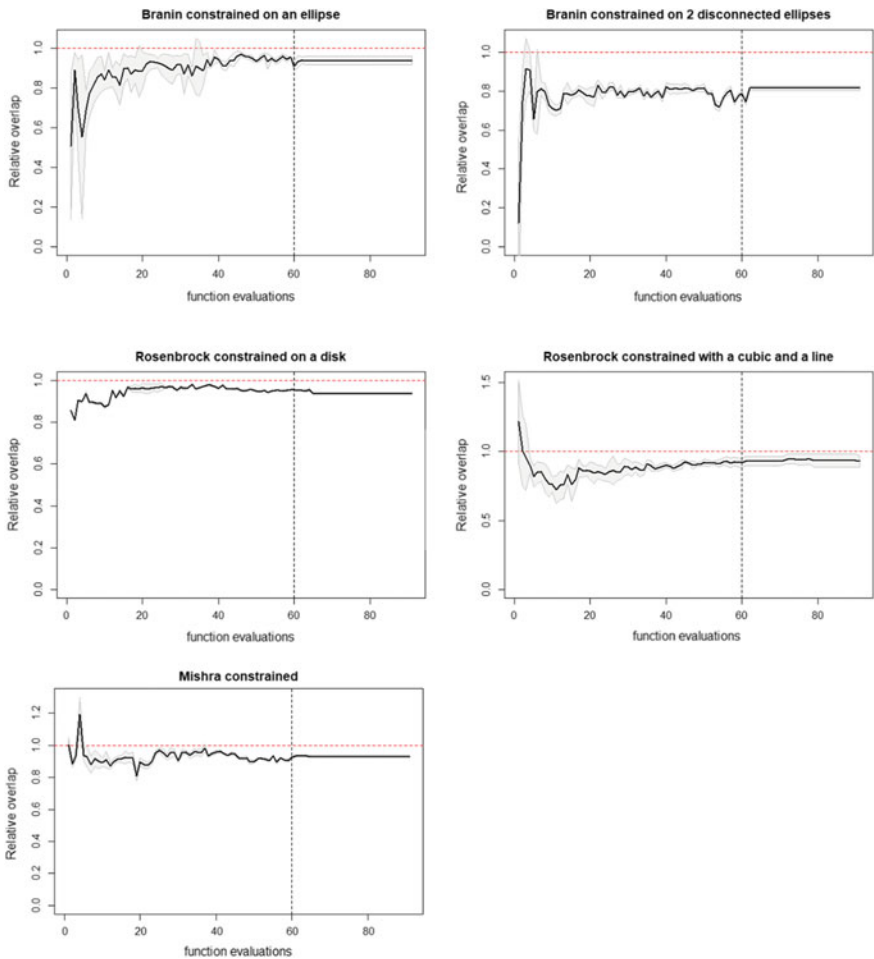


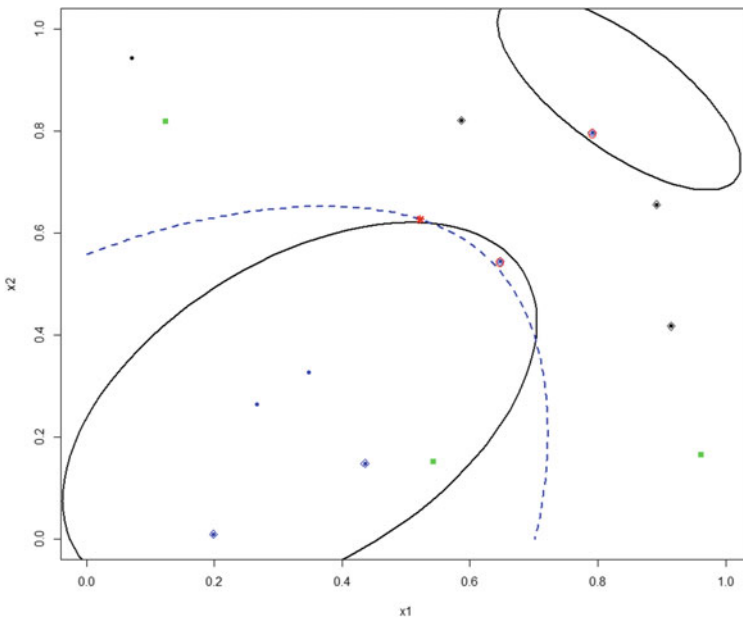
Fig. 5.2 Overlap metric measuring the quality of feasibility estimation provided by the proposed approach, for each one of the five test functions, respectively

further function evaluations outside the actual feasible region. Other important considerations relative to significant reduction in computational costs are reported in Candelieri (2019).

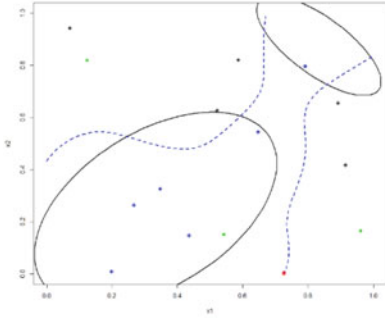
In the following, some explicative pictures relative to how SVM-CBO works are reported, considering a 2D test function (i.e. Branin) where the unknown feasible region consists of the inner areas within two disconnected ellipses.

Begin of phase 1: first estimation of the feasible region according to 10 functions evaluations (i.e. initialization through LHS). The following symbols have been used:

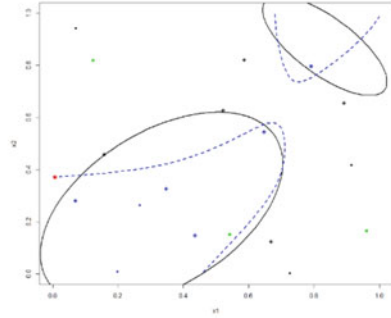
- Solid black line: boundaries of the feasible region (feasible points are inside the ellipses)
- Dotted blue line: boundaries of the estimated feasible region
- Green points: optima of (original) Branin’s test function; only one of them is feasible!
- Blue/black points: evaluations resulting feasible/infeasible
- Points within diamonds—black or blue: support vectors of the two classes
- Red-circled points—black or blue: classification errors with respect to the two classes
- Red asterisk: next point to evaluate.



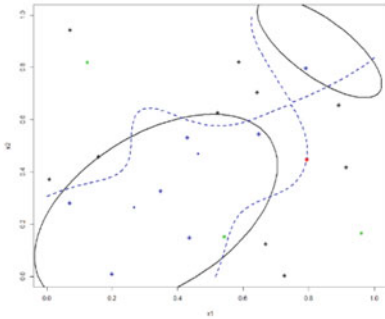
Following, modifications of the estimated feasible region along with some iterations (i.e. 1, 5, 9 and 10) of phase 1:



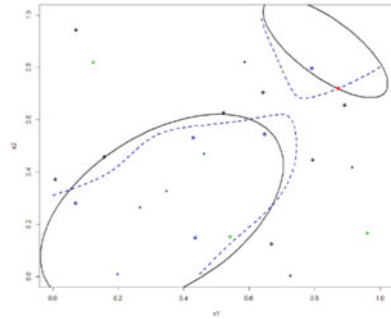
Phase 1 - Iteration #1



Phase 1 - Iteration #5

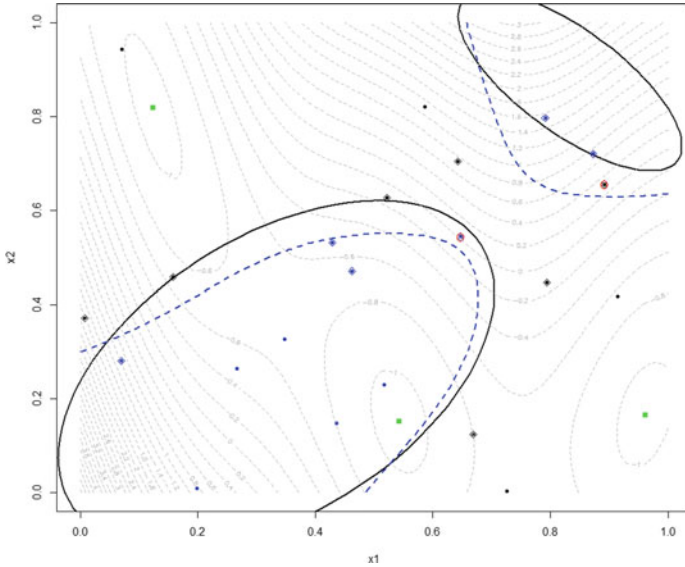


Phase 1 - Iteration #9

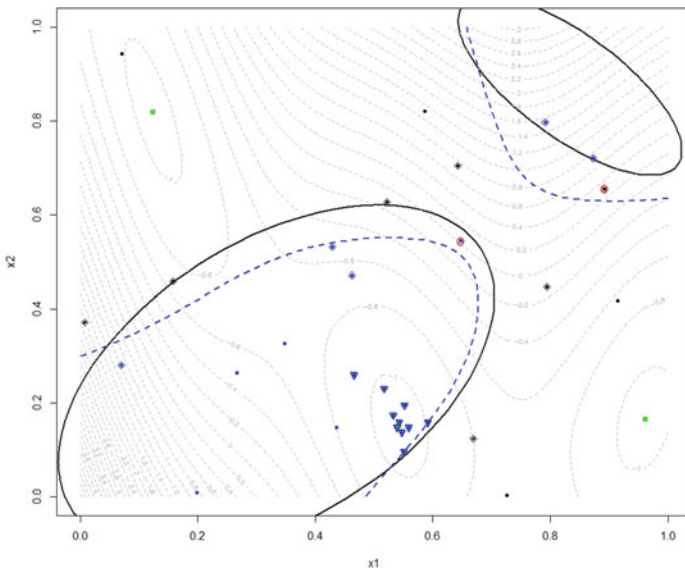


Phase 1 - Iteration #10

End of Phase 1 and beginning of Phase 2: level sets for the original box-constrained objective function are also depicted. In the following figure, the estimated and actual feasible regions are reported.



End of Phase 2 (10 function evaluations more): blue triangles are function evaluations performed in phase 2. All the triangles are in the estimation of the feasible region obtained at the end of phase 1. This means that the estimated feasible region has not been updated during phase 2. The feasible global optimum has been identified in phase 2.



SVM-CBO has some limitations which can be relevant in some application context. First, as any other constrained global optimization approach, it is not well-suited

for dealing with problems where infeasible evaluations are “destructive”. This means that cannot be applied—as is—for solving optimal (online) control of complex industrial systems. This specific kind of problems, known as safe exploration–optimization, will be addressed in the following section. Therefore, simulation–optimization is the setting which might largely benefit from SVM-CBO, such as demonstrated in the case of pump scheduling optimization in water distribution network by using the hydraulic simulation software EPANET 2.0. Secondly, since the SVM classifier models the overall boundary of the feasible region, instead of each individual constraint, a sensitivity analysis cannot be performed. So, the advantage—not only computational—of using one SVM could be partially offset, in the case, we are facing a problem requiring sensitivity analysis.

5.3 Safe Bayesian Optimization

The important difference with respect to CGO is that SafeOpt does not allow—at least with a given probability—any function evaluations outside the feasible region. Moreover, in SafeOpt the objective function is not necessarily partially defined: indeed, a function evaluation is unsafe if the corresponding value of the objective function violates the safety threshold.

It is important to highlight that: contrary to the rest of this book, where the global optimization problem is defined as a minimization problem, for safe optimization we consider a maximization problem, to be coherent with the literature on this topic. Thus, we want to maximize some black-box expensive function without performing function evaluations under a given value.

An approach is presented in Sui et al. (2015), which tries to optimize taking into account the—supposedly known—Lipschitz constant of the objective function. An extension of this approach is presented in Berkenkamp et al. (2016), where the SafeOpt algorithm is applied on high-dimensional problems and without using the information on Lipschitz constant.

The basic problem was stated as: $\max_{x \in X} f(x)$, subject to the safety constraint $f(x) \geq h$, where h is a safety threshold.

A new algorithm, namely StageOpt, has been proposed in Sui et al. (2018), where the SafeOpt algorithm has been generalized and extended to be more efficient and applicable to a broader class of problems. The goal is to optimize a black-box objective function $f : D \rightarrow \mathbb{R}$ from noisy evaluations at the sample points $x_1, x_2, \dots, x_n \in X$. Any point in the sequence, when sampled, must be “safe”, which means that for each one of m unknown safety functions $g_i(x) : X \rightarrow \mathbb{R}$ it lies above some safety threshold $h_i \in \mathbb{R}$.

This optimization problem is formulated as $\max_{x \in X} f(x)$ subject to the safety constraints $g_i(x) \geq h_i$ for $i = 1, \dots, m$.

This is a more general formalization of the original SafeOpt (Sui et al. 2015), which used only a simple threshold on the value of the objective function, leading to a single constraint $g(x) = f(x) \geq h$.

As in constrained optimization, GPs are used to model both the objective function and the constraints. An important assumption in StageOpt is that each safety function g_i is L_i -Lipschitz continuous, with respect to some metric on X . This assumption is usually satisfied by the most commonly used kernels (Srinivas et al. 2010; Sui et al. 2015). Additionally, at least one initial “seed” set of safe points must be given, denoted as $S_0 \subset X$.

Since safe optimization works by safely expanding S_0 , it is not guaranteed to identify the global optimizer x^* , specifically in the case that the region around x^* is disconnected from S_0 . The key component of SafeOpt is the *one-step reachability operator*:

$$R_\varepsilon(S) := S \cup \bigcap_i \{x \in X \mid \exists x' \in S, g_i(x') - \varepsilon - L_i d(x', x) \geq h_i\}$$

where S is the current set of safe points (initially S_0), d is a distance measure in \mathbb{R}^d and ε is the absolute error in considering x as safe. This operator allows to identify the set of all the points estimated as safe depending on the previous evaluations of f on S . Then, given the maximum number of function evaluations, N , we can define the subset of X reachable after N iterations from the initial safe seed set S_0 as the following: $R_\varepsilon^N(S_0) := R_\varepsilon(R_\varepsilon \dots (R_\varepsilon(S_0)) \dots)$, N times. Thus, the optimization problem becomes:

$$x^* = \operatorname{argmax}_{x \in R_\varepsilon^N(S_0)} f(x).$$

StageOpt separates the safe optimization problem into two stages: an exploration phase in which the safe region is iteratively expanded, followed by an optimization phase in which Bayesian optimization is applied within the safe region. Similarly to SVM-CBO, presented for constrained BO, the overall number of function evaluations, N is divided into N_1 for phase 1 and N_2 for phase 2, with $N = N_1 + N_2$. To map the uncertainty of the GP model at the generic iteration n , StageOpt used the confidence intervals:

$$Q_n^i(x) := [\mu_n^i(x) \pm \beta_n \sigma_n^i(x)]$$

where β_n defines the level of confidence. It is easy to note that they correspond to the upper and lower confidence bound of the GP. In the formula, superscripts index the corresponding safety functions, while subscripts index iterations, as usual. Then, to guarantee both safety and progress in safe region expansion, StageOpt uses the following confidence intervals $C_{n+1}^i(x) := C_n^i(x) \cap Q_n^i(x)$, with $C_0^i(x) := [h_i, \infty]$ so that C_{n+1}^i are sequentially contained in C_n^i for all $n = 0, \dots, N$. Upper and lower bounds of C_n^i can be computed and denoted as u_n^i and l_n^i , respectively.

The first stage of StageOpt is safe region expansion. An increasing sequence of safe subsets $S_n \subseteq X$ is computed based on the confidence intervals of the GP posterior:

$$S_{n+1} = \bigcap_i \bigcup_{x \in S_n} \{x' \in X | l_{n+1}^i(x) - L_i d(x, x') \geq h_i\}$$

At each iteration, StageOpt computes a set of expanders points G_n while definition is based on the function:

$$e_n(x) := \left| \bigcap_i \{x' \in X \setminus S_n | u_n^i(x) - L_i d(x, x') \geq h_i\} \right|$$

which (optimistically) quantifies the potential enlargement of the current safe set after sample a new decision x . Then, G_n is defined as follows:

$$G_n = \{x \in S_n : e_n(x) > 0\}$$

Finally, at each iteration StageOpt selects the expander with the highest predictive uncertainty, given by $x_{n+1} = \operatorname{argmax}_{x \in G_n} u_n^i - l_n^i$.

The second stage of StageOpt is BO applied within the safe region identified at the end of the first stage. GP-UCB is used as acquisition function.

In Sui et al. (2018) is illustrated a graphical comparison of the behaviour of SafeOpt (Berkenkamp et al. 2016; Sui et al. 2015) and StageOpt starting from the same safe seed point. The figure is reported as follows (Fig. 5.3).

Initially, both algorithms select the same points, as they use the same definition of safe expansion. However, StageOpt selects noticeably better optimization points than SafeOpt due to the UCB criterion.

In real applications, it may be difficult to compute an accurate estimate of the Lipschitz constants, which may have an adverse effect on the definition of the safe region and its expansion dynamics. One might use one of the solutions for updating L , outlined in Chap. 2, like DIRECT or MultL (Sergeyev and Kvasov 2017).

Within the safe optimization framework, one can use a modified version of SafeOpt that defines safe points using only the GPs (Berkenkamp et al. 2016). This modification can be directly applied to StageOpt as well. This alternative algorithm is structured as follows:

1. a point \bar{x} is considered safe if the lower confidence bound of each safety GPs lies above the respective threshold, that is $l_i(\bar{x}) > h_i$
2. a safe point x' is considered as an expander depending on an optimistic measure of the expansion from x' . More practically, for each constraint a “temporary” GP is trained on the observations performed so far and an unsafe point x which represents an “artificial” evaluation: the associated value of the constraint function

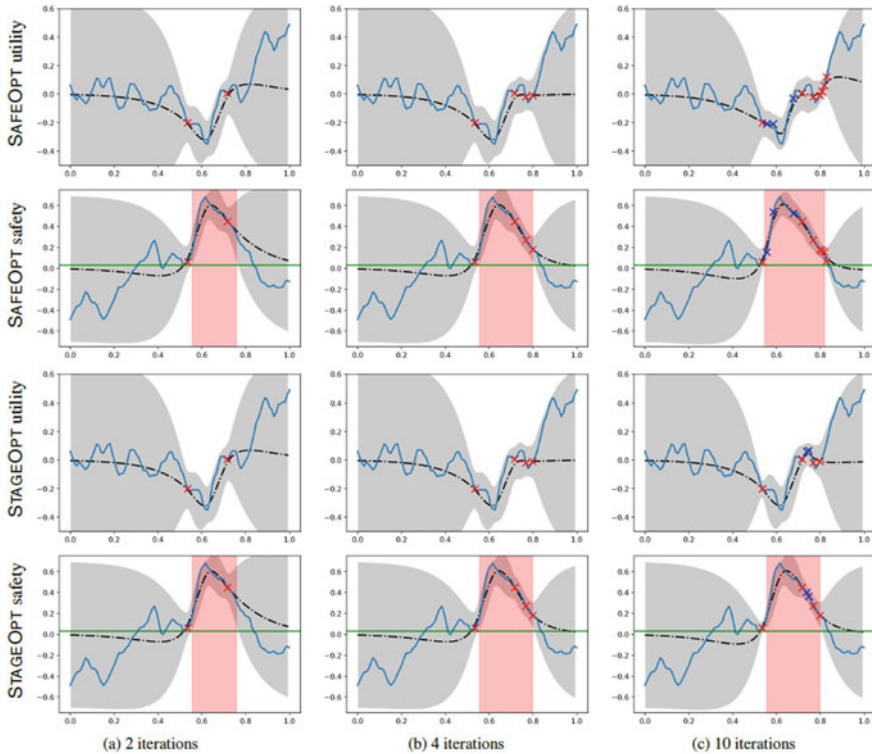


Fig. 5.3 Evolution of GPs in SafeOpt and StageOpt for a fixed safe seed; dashed lines correspond to the mean and shaded areas to ± 2 standard deviations. The first and third rows depict the utility function, and the second and fourth rows depict a single safety function. The utility and safety functions were randomly sampled from a zero-mean GP with a Matern kernel and are represented with solid blue lines. The safety threshold is shown as the green line, and safe regions are shown in red. The red markers correspond to safe expansions and blue markers to maximizations and optimizations. We see that StageOpt identifies actions with higher utility than SafeOpt. (Source: Sui et al. 2018)

in x is given by the upper confidence bound of the “current” GP (which is trained, contrary to the “temporary” one, only on the observations performed so far). If the lower bound of all the “temporary” GPs, computed at x is greater than the associated safety threshold, then x results in a previously unsafe point which can be considered safe with respect to the optimistic expansion from x' . Safe points x' that provide an optimistic expansion are considered as expanders.

We must remark that nobody, at our knowledge, has considered the safe optimization problem subject to both the “usual” constraints $c_i(x)$ and the safety constraints $g_i(x)$. The main difficulty in this challenge is that evaluating a point that is unsafe - in SafeOpt - implies the termination of the optimization process while evaluating an infeasible point - in CGO - only requires to update the estimate of the feasible region.

5.4 Parallel Bayesian Optimization

Performing function evaluations in parallel is a conceptually appealing way to solve optimization problems in less time that would be required by a sequential approach. Some global optimization methods, like Pure Random Search, are “embarrassingly parallel”: this is not the case of BO, which is an intrinsically sequential process, so that its parallelization raises important design issues.

The main distinction between the parallel methods proposed in literature are between *synchronous* and *asynchronous* approaches. With respect to the first class of methods, a multi-point version of EI, called q -EI has been proposed in Ginsbourger et al. (2008), defined as:

$$q\text{EI}(x_{n+1}, \dots, x_{n+q}) = \mathbb{E} \left[\max \left\{ \left(\min(\mathbf{y}) - y_{n+1} \right)^+, \dots, \left(\min(\mathbf{y}) - y_{n+q} \right)^+ \right\} \right]$$

where $\mathbf{y} = \{y_1, \dots, y_n\}$ and $(\cdot)^+$ returns 0 if the argument is less than 0, otherwise returns the argument itself. This means to deal with a minimum of dependent random variables: the exact joint distribution of the q unknown function evaluations, y_{n+1}, \dots, y_{n+q} , conditioned on previous observations through the current GP is given by:

$$(y_{n+1}, \dots, y_{n+q} | D_{1:n}) \sim \mathcal{N}((\mu(x_{n+1}), \dots, \mu(x_{n+q})), \Sigma_q)$$

where Σ_q is the covariance matrix updated with the q points x_{n+1}, \dots, x_{n+q} .

This synchronous parallelization approach is aimed at measuring the joint potential of a given additional set of q points to evaluate. The main idea is to consider q synchronous processors and to maximize q -EI at each iteration; then the GP model is updated with the entire set of new observations. However, as stated in Ginsbourger et al. (2008), solving the maximization of q -EI can be unaffordable when both the number of dimensions and the number of points q increase.

Computational approaches for parallelizing knowledge gradient (KG) have been also proposed (Wu and Frazier 2016). The parallel KG, namely q -KG, is Bayes-optimal for minimizing the minimum of the predictor of the GP if only one decision is remaining. The q -KG algorithm will reduce to the parallel EI algorithm in the noise-free setting and the final recommendation is restricted to the previous sampled points. The basic issue is that computing q -KG and its gradient is very expensive. Analogously to q -EI, the aim of q -KG is to identify a batch of q points to evaluate, basically in a synchronous setting. Another interesting approach, namely “portfolio allocation”, has been already discussed in Chap. 4.

To address the inefficiency due to the synchronization, a new asynchronous approach, based on EI, is proposed in Ginsbourger et al. (2011). This approach, namely expectation of EI (EEI), focuses on the case of asynchronous parallel BO, where a new function evaluation is performed before all the other processors have produced their result. At every time, observations in $D_{1:n}$ are divided in three dif-

ferent groups: (i) already evaluated, (ii) currently under evaluation (aka “busy”) and (iii) candidate points for forthcoming function evaluations. The main issue is how to take busy points into account within the GP model.

The issue on how to inject partial knowledge from an ongoing evaluation at a busy point into the EI criterion, without knowing the corresponding actual value of the function, is basically solved probabilistically:

$$\text{EEI}(\cdot; x^{\text{busy}}) = \int \text{EI}(\cdot; x^{\text{busy}}, y^{\text{busy}}) \text{pdf}(Y(x^{\text{busy}})|D) y^{\text{busy}} dy^{\text{busy}}$$

where $\text{pdf}(Y(x^{\text{busy}})|D)$ is the probability density function of the random variable $Y(x^{\text{busy}})$ conditional on the function evaluations performed so far and stored in $D = D_{1:n}$. It is important to highlight that $\text{EI}(\cdot; x^{\text{busy}}, y^{\text{busy}})$ depends on y^{busy} in a quite complicated non-linear way, without any chance to have an analytical expression for EEI, even if $Y(x^{\text{busy}}|D)$ is known and simple. Therefore, EEI is approximated by averaging on a sufficient number of y^{busy} sampled according to the $Y(x^{\text{busy}}|D)$.

A straightforward way of getting statistical estimates of EEI is based on Monte Carlo sampling. The EEI approach relies on conditional simulations: it proved to be a sensible criterion, but practically not straightforward to optimize.

Finally, in Kandasamy et al. (2018) a parallel version of Thompson sampling (TS) is proposed. A theoretical analysis proves that a direct application of the sequential TS algorithm, presented in Chap. 3, in either synchronous or asynchronous parallel settings, offers a powerful result: performing n evaluations distributed among q workers is equivalent to performing n evaluations in sequence. Experimental results on simulation–optimization problems and hyperparameters optimization of a convolutional neural network prove that asynchronous TS outperforms many existing parallel BO algorithms.

5.5 Multi-objective Bayesian Optimization

Multi-objective optimization is characterized by a set of objective functions to be minimized $f_i : X \rightarrow R, i = 1, \dots, m$. The solution to this kind of problem is not unique and defined as $x \in X : \nexists \bar{x} \in X$, such that $f(\bar{x}) \prec f(x)$, where symbol \prec denotes the dominance relation, that is:

$$\forall i f_i(x) \leq f_i(\bar{x}) \text{ and } \exists j \text{ s.t. } f_j(x) < f_j(\bar{x})$$

This allows to identify the Pareto front, defined by the values of different objective functions and the set of corresponding dominant solutions $x \in X$.

In BO, a typical approach is to model every f_i through a Gaussian process, GP_i , with $i = 1 \dots m$. In Emerich and Klinkenberg (2008), the EI acquisition function has been generalized to the multi-objective case by considering the volume of the dominated region: the improvement provided by a new evaluation is the increase in

the dominated volume. This extension of EI to the multi-objective case is also known as expected hyper-volume improvement (EHVI).

Given the n function evaluations performed so far, with values $f(x_i) = (f_1(x_i), \dots, f_m(x_i))$, $i = 1, \dots, n$, we define the set $H_n = \{y \in \mathbb{B}, \exists i \leq n, f(x_i) < y\}$, where $\mathbb{B} \subset \mathbb{R}^m$ is $\mathbb{B} = \{y \in \mathbb{R}^m; y \leq \bar{y}\}$ with $\bar{y} \in \mathbb{R}^m$ is introduced to guarantee that H_n is finite. Thus, H_n is the subset of \mathbb{B} whose points are dominated by the previous evaluations.

Finally, the improvement provided by a new evaluation can be measured as: $I(x_{n+1}) = |H_{n+1}| - |H_n|$, where $|\cdot|$ denotes the volume in \mathbb{R}^m .

Since $H_{n+1} \supset H_n$, the value $I(x_{n+1})$ is the increase in the volume of the dominated region (Fig. 5.4).

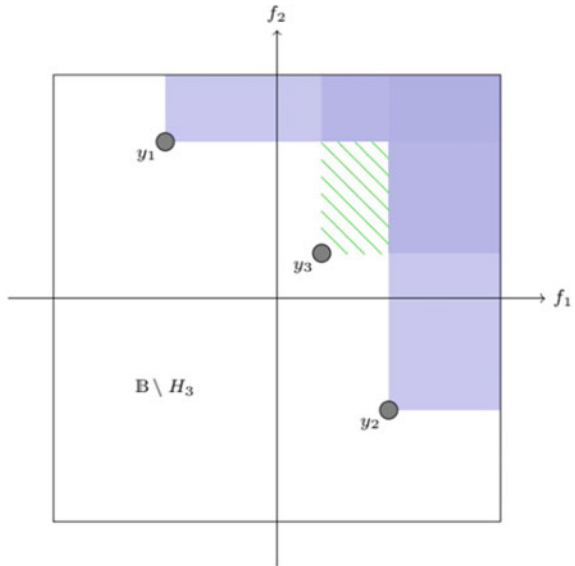
Finally, maximizing EHVI means maximizing the expected improvement on $I(x^{n+1})$ and it can be analytically defined as follows:

$$EHVI(x) = \int_{\mathbb{B} \setminus H_n} \mathbb{P}_n(\xi(x) < y) dy$$

where \mathbb{P}_n is the probability conditioned on $x_1, \xi(x_1), \dots, x_n, \xi(x_n)$, with $\xi = (\xi_1, \dots, \xi_m)$ is the vector of GPs modelling the corresponding objective functions $f = (f_1, \dots, f_m)$.

In Feliot et al. (2017), the EHVI has been extended to the constrained multi-objective optimization case and to the case in which the user expresses preference-order constraints (Abdolshah et al. 2019) The EHVI acquisition function is now

Fig. 5.4 Example of an improvement of the dominated region. The regions dominated by y_1 and y_2 are represented in shaded areas, with darker shades indicating overlapping regions. The hatched area corresponds to the improvement of the dominated region resulting from the observation of y_3 . (Source Feliot 2017)



considered as the dominant approach and some implementations are available in Matlab and the *R* packages “Gpareto” and “mlrMBO” (Horn 2015).

The EHVI acquisition function has been also extended in Wada and Hinu (2019) to deal with multi-point search.

An interesting application of multi-objective optimization to BO has been proposed in Calvin and Zilinskas (2019), where the acquisition of the next point to evaluate is defined as a bi-objective optimization problem. In this paper, it is shown that two well-known acquisition function, EI and PI, can be framed into the bi-objective approach, whose the objectives are $\mu(x)$ and $\sigma(x)$ of a probabilistic surrogate model.

5.6 Multi-source and Multi-fidelity Bayesian Optimization

Function evaluations in simulation-based experiments or black-box optimization are usually expensive. The basic idea in multi-fidelity is that, in the initial stages of the optimization, “cheaper” evaluation of the objective function might yield anyway considerable information while reducing the computational cost. In this framework, one is considering a collection of information sources, denoted by $f(x, s)$, where s represents a specific level of “fidelity” of the source. Usually, the lower s the higher the fidelity, with $f(x, 0)$ corresponding to the original objective function (Frazier 2018). Decreasing the fidelity decreases the accuracy of the approximation on $f(x)$, leading to a reduction in the cost of evaluation but to a poorer accuracy in the estimate of the objective function.

In multi-fidelity, the aim is $\min_{x \in X} f(x)$ but observing $f(x, s)$ at a sequence of points and fidelities $\{(x_i, y_i)\}_{1:n}$ with a total cost lower than a given budget, that is $\sum_{i=1}^n c(s_n) \leq B$, where $c(s_n)$ is the cost to evaluate at fidelity s_n and B is the given available budget.

Acquisition functions may present some limitations in the development of multi-fidelity BO. Indeed, while KG, ES and PES can be applied, as shown in Poloczek (2017), EI is useless because evaluating $f(x, s)$ for $s \neq 0$ never offers an improvement in the best seen (i.e. $EI = 0$ for $s \neq 0$), leading to the selection of the highest fidelity source only (Frazier 2018).

Multi-fidelity optimization has been also considered for the optimization of machine learning algorithms (Klein et al. 2017; Kandasamy et al. 2019). The problem of hyperparameter tuning is also considered in Sen et al. (2018) which proposes to train the algorithm on a subsampled version of the whole dataset.

Multi-fidelity optimization has also been gaining a growing importance in several applications also in analogue circuit synthesis (Zhang et al. 2019) and engineering design (Linz et al. 2017) who propose an optimization algorithm that guides the search for solutions on a high-fidelity model through the approximation of a level set from a low-fidelity model. Using the probabilistic branch-and-bound algorithm to approximate a level set for the low-fidelity model, they are able to efficiently locate

solutions inside of a target quantile, and therefore reduce the number of high-fidelity evaluations needed in searches.

An interesting application of multi-fidelity BO is presented in Perdikaris et al. (2016) for model inversion in hemodynamics and biomedical engineering (Costabal et al. 2019). A comprehensive multi-fidelity framework, in the context of bandit problems, has been proposed in Kandasamy et al. (2017) where the objective function and its approximations are sampled from a GP. Another interesting work is reported in Ghoreishi and Allaire (2018) which proposes a multi-source information approach for the constrained case.

References

- Abdolshah, M., Shilton, A., Rana, S., Gupta, S., Venkatesh, S.: Multi-objective Bayesian optimisation with preferences over objectives (2019). arXiv preprint [arXiv:1902.04228](https://arxiv.org/abs/1902.04228)
- Bachoc, F., Helbert, C., Picheny, V.: Gaussian process optimization with failures: classification and convergence proof (2019)
- Basudhar, A., Dribusch, C., Lacaze, S., Missoum, S.: Constrained efficient global optimization with support vector machines. *Struct. Multidisciplinary Optim.* **46**, 201–221 (2012). <https://doi.org/10.1007/s00158-011-0745-5>
- Berkenkamp, F., Krause, A., Schoellig, A.P.: Bayesian optimization with safety constraints: safe and automatic parameter tuning in robotics (2016). arXiv preprint [arXiv:1602.04450](https://arxiv.org/abs/1602.04450). doi:10.1177
- Calvin, J.M., Žilinskas, A.: On efficiency of bicriteria optimization. In: AIP Conference Proceedings, vol. 2070, no. 1, p. 020035. AIP Publishing (2019, February)
- Candelieri, A., Archetti, F.: Sequential model-based optimization with black-box constraints: feasibility determination via machine learning. In: AIP Conference Proceedings. p. 020010 (2019)
- Candelieri, A., Galuzzi, B.G., Giordani, I., Perego, R., Archetti, F.: Optimizing partially defined black-box functions under unknown constraints via Sequential Model Based Optimization: an application to Pump Scheduling Optimization in Water Distribution Networks. To appear in Proceedings of Learning and Intelligent Optimization conference (LION 13) (2019)
- Candelieri, A., Giordani, I., Archetti, F., Barkalov, K., Meyerov, I., Polovinkin, A., Sysoyev, A., Zolotykh, N.: Tuning hyperparameters of a SVM-based water demand forecasting system through parallel global optimization. *Comput Oper Res* (2018). <https://doi.org/10.1016/j.cor.2018.01.013>
- Costabal, F.S., Perdikaris, P., Kuhl, E., Hurtado, D.E.: Multi-fidelity classification using Gaussian processes: accelerating the prediction of large-scale computational models (2019). arXiv preprint [arXiv:1905.03406](https://arxiv.org/abs/1905.03406)
- Digabel, S.L., Wild, S.M.: A taxonomy of constraints in simulation-based optimization (2015). arXiv preprint [arXiv:1505.07881](https://arxiv.org/abs/1505.07881)
- Emmerich, M., Klinkenberg, J.W.: The computation of the expected improvement in dominated hypervolume of Pareto front approximations. *Rapport technique*, Leiden University, 34, 7–3 (2008)
- Feliot, P., Bect, J., Vazquez, E.: A Bayesian approach to constrained single-and multi-objective optimization. *J. Global Optim.* **67**(1–2), 97–133 (2017)
- Frazier, P.I.: Bayesian optimization. In: *Recent Advances in Optimization and Modeling of Contemporary Problems*, pp. 255–278. INFORMS (2018)
- Gardner, J.R., Kusner, M.J., Xu, Z.E., Weinberger, K.Q., Cunningham, J.P.: Bayesian Optimization with Inequality Constraints Jacob. In: *ICML*, pp. 937–945 (2014)
- Ghoreishi, S.F., Allaire, D.: Multi-information source constrained Bayesian optimization. *Struct. Multidisciplinary Optim.* 1–15 (2018). <https://doi.org/10.1007/s00158-018-2115-z>

- Ginsbourger, D., Riche, R. Le, Carraro, L.: A multi-points criterion for deterministic parallel global optimization based on Gaussian processes. In: International Conference on Nonconvex Programming, NCP07, Rouen, France. 1–30 (2008)
- Ginsbourger, D., Riche, R. Le: Dealing with asynchronicity in parallel Gaussian Process based global optimization *. In: 4th International Conference of the ERCIM WG on Computing & Statistics (ERCIM'11) (2011)
- Gramacy, R.B., Lee, H.K.H.: Optimization under unknown constraints. *Bayesian Statistics*, 9 (2011)
- Gramacy, R.B., Gray, G.A., Le Digabel, S., Lee, H.K., Ranjan, P., Wells, G., Wild, S.M.: Modeling an augmented Lagrangian for blackbox constrained optimization. *Technometrics* **58**(1), 1–11 (2016)
- Hernández-Lobato, J.M., Gelbart, M.A., Hoffman, M.W., Adams, R.P., Ghahramani, Z.: Predictive entropy search for bayesian optimization with unknown constraints (2015)
- Horn, D., Wagner, T., Biermann, D., Weihs, C., Bischl, B.: Model-based multi-objective optimization: Taxonomy, multi-point proposal, toolbox and benchmark. In: International Conference on Evolutionary Multi-Criterion Optimization, pp. 64–78. Springer, Cham (2015, March)
- Jones, D.R., Schonlau, M., Welch, W.J.: Efficient global optimization of expensive black-box functions. *J. Glob. Optim.* **13**(4), 455–492 (1998)
- Kandasamy, K., Dasarathy, G., Schneider, J., Póczos, B.: Multi-fidelity bayesian optimisation with continuous approximations. In: Proceedings of the 34th International Conference on Machine Learning, vol. 70, pp. 1799–1808. JMLR. org (2017, August)
- Kandasamy, K., Krishnamurthy, A., Schneider, J., Poczso, B.: Asynchronous parallel Bayesian optimisation via Thompson sampling (2018). arXiv preprint [arXiv:1705.09236](https://arxiv.org/abs/1705.09236)
- Kandasamy, K., Neiswanger, W., Schneider, J., Poczso, B., Xing, E.: Neural architecture search with Bayesian optimisation and optimal transport (2019). arXiv preprint [arXiv:1802.07191](https://arxiv.org/abs/1802.07191)
- Klein, A., Falkner, S., Bartels, S., Hennig, P., Hutter, F.: Fast Bayesian optimization of machine learning hyperparameters on large datasets. In: Proceedings of the 20th International Conference on Artificial Intelligence and Statistics. Proceedings of Machine Learning Research, Brookline, MA, 54, 528–536 (2017)
- Letham, B., Karrer, B., Ottoni, G., Bakshy, E.: Constrained Bayesian optimization with noisy experiments. *Bayesian Anal.* **14**(2), 495–519 (2019)
- Linz, D.D., Huang, H., Zabinsky, Z.B.: Multi-fidelity simulation optimization with level set approximation using probabilistic branch and bound. In: Winter Simulation Conference (WSC), IEEE, pp. 2057–2068 (2017, December)
- Perdikaris, P., Karniadakis, G.E.: Model inversion via multi-fidelity Bayesian optimization: a new paradigm for parameter estimation in haemodynamics, and beyond. *J. R. Soc. Interface* **13**(118), 20151107 (2016)
- Picheny, V., Gramacy, R. B., Wild, S., Le Digabel, S.: Bayesian optimization under mixed constraints with a slack-variable augmented Lagrangian. In: Advances in Neural Information Processing Systems, pp. 1435–1443 (2016)
- Poloczek, M., Wang, J., Frazier, P. Multi-information source optimization. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) *Advances in Neural Information Processing Systems*, 30, pp. 4291–4301. Curran Associates, Red Hook, NY, (2017)
- Rudenko, L.I.: Objective functional approximation in a partially defined optimization problem. *J. Math. Sci.* **72**(5), 3359–3363 (1994)
- Sacher, M., Duveigneau, R., Le Maitre, O., Durand, M., Berrini, E., Hauville, F., Astolfi, J.A.: A classification approach to efficient global optimization in presence of non-computable domains. *Struct. Multidisciplinary Optim.* **58**(4), 1537–1557 (2018)
- Sen, R., Kandasamy, K., Shakkottai, S.: Multi-fidelity black-box optimization with hierarchical partitions. In: International Conference on Machine Learning, pp. 4545–4554 (2018, July)
- Sergeyev, Y.D., Kvasov, D.E.: *Deterministic Global Optimization: an Introduction to the Di-agonal Approach*. Springer (2017)
- Sergeyev, Y.D., Kvasov, D.E., Khalaf, F.M.: A one-dimensional local tuning algorithm for solving GO problems with partially defined constraints. *Optim. Lett.* **1**(1), 85–99 (2007)

- Sui, Y., Gotovos, A., Burdick, J.W., Krause, A.: Safe exploration for optimization with gaussian processes. In: International Conference on Machine Learning (ICML), (SafeOpt) (2015)
- Sui, Y., Zhuang, V., Burdick, J.W., Yue, Y.: Stagewise safe Bayesian optimization with Gaussian processes (2018). arXiv preprint [arXiv:1806.07555](https://arxiv.org/abs/1806.07555)
- Tsai, Y.A., Pedrielli, G., Mathesen, L., Zabinsky, Z.B., Huang, H., Candelieri, A., Perego, R.: Stochastic optimization for feasibility determination: an application to water pump operation in water distribution network. In: Proceedings of the 2018 Winter Simulation Conference, pp. 1945–1956. IEEE Press (2018, December)
- Vapnik, V.: Statistical Learning Theory. Willey, New York (1998)
- Wada, T., Hino, H.: Bayesian optimization for multi-objective optimization and multi-point search (2019). arXiv preprint [arXiv:1905.02370](https://arxiv.org/abs/1905.02370)
- Wu, J., Frazier, P.: The parallel knowledge gradient method for batch bayesian optimization. In: Advances in Neural Information Processing Systems, pp. 3126–3134 (2016)
- Zhang, S., Lyu, W., Yang, F., Yan, C., Zhou, D., Zeng, X., Hu, X.: An efficient multi-fidelity Bayesian optimization approach for analog circuit synthesis. In: DAC, pp. 64–1 (2019, June)