

A Classification Approach to Efficient Global Optimization in Presence of Non-Computable Domains

Matthieu Sacher^{a,*}, Régis Duvigneau^b, Olivier Le Maître^c, Mathieu Durand^d,
Élisa Berrini^e, Frédéric Hauville^a, Jacques-André Astolfi^a

^aNaval Academy Research Institute - IRENAV, France

^bUniversité Côte d'Azur, INRIA - CNRS, France

^cLIMSI - CNRS, Université Paris Saclay, France

^dK-EPSILON & Groupama Team France & Sirlu Innovation, France

^eUniversité Côte d'Azur, INRIA - CNRS & MyCFD, France

Abstract

Gaussian-Process based optimization methods have become very popular in recent years for the global optimization of complex systems with high computational costs. These methods rely on the sequential construction of a statistical surrogate model, using a training set of computed objective function values, which is refined according to a prescribed infilling strategy. However, this sequential optimization procedure can stop prematurely if the objective function cannot be computed at a proposed point. Such a situation can occur when the search space encompasses design points corresponding to an unphysical configuration, an ill-posed problem, or a non-computable problem due to the limitation of numerical solvers. To avoid such a premature stop in the optimization procedure, we propose to use a classification model to learn non-computable areas and to adapt the infilling strategy accordingly. Specifically, the proposed method splits the training set into two subsets composed of computable and non-computable points. A surrogate model for the objective function is built using the training set of computable points, only, whereas a probabilistic classification model is built using the union of the computable and non-computable training sets. The classifier is then incorporated in the surrogate-based optimization procedure to avoid proposing new points in the non-computable domain while improving the classification uncertainty if needed. The method has the advantage to automatically adapt both the surrogate of the objective function and the classifier during the iterative optimization process. Therefore, non-computable areas do not need to be *a priori* known. The proposed method is applied to several analytical problems presenting different types of difficulty, and to the optimization of a fully nonlinear fluid-structure interaction system. The latter problem concerns the drag minimization of a flexible hydrofoil with cavitation constraints. The efficiency of the proposed method compared favorably to a reference evolutionary algorithm, except for situations where the feasible domain is a small portion of the design space.

Keywords: Global optimization, Gaussian process model, constrained optimization, classification

1. Introduction

The use of surrogate models is a classical approach to reduce the computational burden related to the optimization of complex systems [1]. Polynomial models were first used as surrogates, thanks to their ease of construction. More recently, sophisticated surrogate models such as Gaussian processes (GP) [2] or Support Vector Machines (SVM) [3] have emerged. Gaussian Process (GP) modeling is a statistical method to approximate functions from a finite set of observations, possibly noisy, at arbitrary points. Using the

*Corresponding author

Email address: matthieu.sacher@ecole-navale.fr (Matthieu Sacher)

observations, it explicitly updates the prior mean and covariance functions into their posterior counterparts, minimizing the mean-squared-error of the estimator. In the case of Gaussian observation noise, the estimator corresponds to the optimal Bayesian posterior. Contrary to alternative regression-type approaches, GP modeling does not require to prescribe explicitly a basis for the approximation space. GP surrogates have been applied to several problems such as uncertainty quantification [4, 5], binary classification [6], multi-levels system [7], multi-fidelity surrogate [8] or clustering-based space exploration [9].

GP models have also been found especially appealing for optimization, in the framework of the Efficient Global Optimization (EGO) [10] method, because their statistical nature allows to provide both a prediction of the objective function, in terms of model mean, and an error estimate, in terms of model variance. EGO strategies have been used in several engineering applications, such as aerodynamic drag reduction of transonic wings [11], vibration reduction for rotating aircrafts [12, 13], optimization of FSI problems [14] and sail trimming optimization [15]. The EGO efficiency has been demonstrated for the optimization of complex systems with costly objective function evaluations [1].

In all these problems, a statistical criterion accounting for the Expected Improvement (EI) [16], usually referred to as *merit function*, is used to select a new (or multiple [17]) design point, at which the objective function should be evaluated. This new evaluation is used to improve the accuracy of the surrogate. However, in some situations, the analysis may not be possible for the newly selected design point. In the context of simulation-based optimization, such situations can be related for instance to the existence of unphysical configurations, ill-posed problems, or the lack of numerical robustness (weak convergence of the solvers, non-convergence to a steady solution, unstable computation, poor mesh quality for extreme configuration, etc). The origin of such failures is dependent on the application domain and an example will be provided below in the context of hydrodynamic design. Note that the automatic detection of non-computable points is a real issue in practical applications. Indeed, the origins of the failures can be very diverse and their detection is therefore not obvious. In some cases, the failure can be relatively easy to detect by monitoring the evolution of residuals or any variable of interest (*e.g.* forces). In other cases, the computational time might be a way to identify the simulations that are not converging. The use of clustering approaches [18] may also be an alternative to detect such computational failures. All these situations will be referred generically as cases of *non-computable design points*. The occurrence of such non-computable points is critical to sequential EGO method because it leads to a premature halt of the optimization process, due to the impossibility to update the surrogate model. A possible way to overcome this problem would be to interpolate/extrapolate the non-computable objective function values, using directly the current GP model or more sophisticated methods [19]. However, the accuracy of these values and their impact on the convergence of the optimization procedure are questionable. Moreover, such methods are not always satisfactory since we may not be interested in non-computable points that correspond to unphysical designs.

Unphysical or ill-posed situations typically arise from the definition of the design space that is too large and includes non-feasible situations. Similarly, the presence of design points that are not computable because of numerical issues should be traced back to a design space that is too rich and contains extreme situations that should not be considered as potential candidates for the optimum. Otherwise, it implies that the numerical solvers considered for the evaluation of the objective function are simply not suitable for the optimization task. However, even though the numerically critical areas of the design space may not be close to the sought optimum, the EGO eventually visit them because of the infilling strategy. A naive fix to the encounter of a non-computable point consists in associating to it a prescribed high (resp. low) value for the objective function to be minimized (resp. maximized). This avoids halting the EGO procedure and prevents the future exploration of the non-computable point neighborhood. Unfortunately, the selection of an appropriate value for the non-computable points is not a trivial task with, in practice, detrimental consequences such as a significant loss of accuracy on the whole GP model approximating the objective function and a possibly strong deterioration of the EGO convergence. Alternative remedies are a better definition of the design space, to explicitly remove design points leading to unphysical or ill-posed problems, and the improvement of the numerical robustness through, *e.g.*, the tuning of solvers parameters. Unfortunately, depending on the situation these two tasks can be extremely cumbersome, if possible at all.

In this work, we propose to combine a classical EGO method and a classification approach in order to deal with the possible existence of non-computable design areas. In this approach, the subset of the

design space corresponding to computable points is progressively determined, by the Least-Squares Support Vector Machine (LS-SVM) classification method [20], on the basis of previously classified computable / non-computable points. The SVM classification method was first proposed in the EGO context by [21] for the treatment of discontinuous or binary constraints. We extend this idea to the case of non-computable design areas, with the suggestion of using LS-SVM instead of SVM models. Indeed, LS-SVMs are computable at a lower cost than SVMs since the SVM formulates the classifier as a quadratic programming problem while LS-SVM deals with a set of linear equations. Specifically, each iteration step in the iterative method for solving the SVM consists of a linear system to be solved which has the same complexity as one single LS-SVM. Moreover, benchmarks in [22] have shown that LS-SVMs are performing consistently very well in comparison with many other methods. The LS-SVM classifier provides a probability that a new point is computable. We incorporate this probabilistic information in the definition of the merit function when selecting the new design point to be analyzed. The merit function also involves a classical EI criterion, based on the GP model of the objective function constructed on the subset of computable design points. Therefore, the key advantages of the proposed method, called EGO-LS-SVM hereafter, are the following: it does not halt when a non-computable point is proposed; it does not require an explicit definition of the computable domain, but rather it constructs it; the GP model of the objective function uses only computable values and so is not polluted by arbitrary values at non-computable points. Finally, the EGO-LS-SVM method can handle optimization problems with inequality constraints and non-computable points, in a unified fashion employing the same classification strategy.

The present paper is organized as follows. The EGO-LS-SVM method is progressively introduced in Section 2, starting with the classical EGO method, presenting next the LS-SVM classification method to handle non-computable domains and inequality constraints, and finally deriving several merit functions. The efficiency of the EGO-LS-SVM method is first assessed on a series of analytical optimization problems in Section 3. A comparison with the reference evolutionary algorithm CMA-ES [23] is also provided. The optimization of a fully nonlinear Fluid-Structure Interaction (FSI) system involving non-computable domains is subsequently considered in Section 4. It consists in the minimization of the hydrodynamic drag of a two-dimensional flexible hydrofoil, with non-cavitation constraints, for a 5-dimensional design space defining the unloaded geometry and mechanical properties of the hydrofoil. Finally, the conclusions of the work are proposed in Section 5.

2. GP-based constrained optimization

Our objective is to estimate the solution \mathbf{x}_{opt} of the following generic optimization problem

$$\mathbf{x}_{\text{opt}} = \arg \min_{\mathbf{x} \in \Omega} f(\mathbf{x}), \quad \text{s.t. } \mathbf{Q}(\mathbf{x}) \geq \mathbf{0}, \quad (1)$$

where \mathbf{x} is the vector of design parameters, $\Omega \subset \mathbb{R}^d$ is the optimization domain, $f : \Omega \mapsto \mathbb{R}$ the objective function and $\mathbf{Q} : \mathbf{x} \mapsto \mathbb{R}^m$ is the vector of (non-linear) constraints. Even in the unconstrained case, $m = 0$, finding the global optimum of f can be very costly, in particular when its evaluation is numerically expensive. The use of surrogate models in place of f is then a classical approach to reduce the computational burden related to the optimization of complex systems [1]. In this work, we consider the use of Gaussian processes (GP) [2] which, owing to their statistical nature, provide both a prediction of the objective function and a measure of the uncertainty (variance) in the prediction. These features are appealing in optimization, as they can be used to derive rigorous optimization strategies based on the maximization of the Expected Improvement (EI) criterion. These methods are globally referred to as the Efficient Global Optimization (EGO) [10] methods. A summary of the construction of the GP model for f is provided in Section 2.1, as well as the resulting optimization strategy in the unconstrained case. The classification approach to account for constraints and non-computable subsets of points is considered in Section 2.2.

2.1. Unconstrained EGO using GP

Consider a set of n training inputs points $\mathbf{X}_n = \{\mathbf{x}_1, \dots, \mathbf{x}_n\}$, each associated to a noisy observation y_i of the objective function f . It is assumed that $y_i = f(\mathbf{x}_i) + \varepsilon_i$, where the ε_i are Gaussian measurement noises,

assumed for simplicity independent and identically distributed with variance σ_ϵ^2 . The GP construction considers that $f(\mathbf{x})$ is a realization of a zero-mean multivariate Gaussian process with covariance function C_f . In this work, we consider the multidimensional squared exponential covariance functions defined by

$$C_f(\mathbf{x}, \mathbf{x}'; \Theta) \doteq \theta_1 \prod_{i=1}^d \exp\left(-\frac{(x_i - x'_i)^2}{2l_i^2}\right) + \theta_2, \quad (2)$$

where $\Theta = \{\theta_1, \theta_2, l_1, l_2, \dots, l_d\}$ is a vector of hyper-parameters to be inferred from the observations. Denoting $\mathbf{C}(\Theta) \in \mathbb{R}^{n \times n}$ the covariance matrix with entries $C_{i,j}(\Theta) \doteq C_f(\mathbf{x}_i, \mathbf{x}_j; \Theta)$, $1 \leq i, j \leq n$, the joint Gaussian distribution of the noisy observation vector, $\mathbf{Y}_n = (y_1, \dots, y_n)^\top$, and the predicted observation $y(\mathbf{x})$ is given by

$$\begin{pmatrix} \mathbf{Y}_n \\ y(\mathbf{x}) \end{pmatrix} \sim \mathcal{N}\left(\mathbf{0}, \begin{bmatrix} \mathbf{C} + \sigma_\epsilon^2 \mathbf{I} & \mathbf{k}(\mathbf{x}) \\ \mathbf{k}^\top(\mathbf{x}) & \kappa(\mathbf{x}) + \sigma_\epsilon^2 \end{bmatrix}\right). \quad (3)$$

In (3) we have denoted $\kappa(\mathbf{x}) \doteq C_f(\mathbf{x}, \mathbf{x}; \Theta)$, $\mathbf{k}(\mathbf{x}) \doteq (C_f(\mathbf{x}, \mathbf{x}_1; \Theta) \cdots C_f(\mathbf{x}, \mathbf{x}_n; \Theta))^\top$ and \mathbf{I} the identity matrix of \mathbb{R}^n . From the conditional rules of joint Gaussian distributions [24], the *best* prediction $\hat{f}(\mathbf{x})$ of $f(\mathbf{x})$, *i.e.* the mean of y , and the prediction variance $\hat{\sigma}_f^2(\mathbf{x})$ are given by

$$\hat{f}(\mathbf{x}) = \mathbf{k}^\top(\mathbf{x}) (\mathbf{C}(\Theta) + \sigma_\epsilon^2 \mathbf{I})^{-1} \mathbf{Y}_n, \quad (4)$$

$$\hat{\sigma}_f^2(\mathbf{x}) = \kappa(\mathbf{x}) + \sigma_\epsilon^2 - \mathbf{k}^\top(\mathbf{x}) (\mathbf{C}(\Theta) + \sigma_\epsilon^2 \mathbf{I})^{-1} \mathbf{k}(\mathbf{x}). \quad (5)$$

The hyper-parameters Θ and noise variance σ_ϵ^2 can be determined by maximizing the log-marginal likelihood [24] using the ‘‘Covariance Matrix Adaptation Evolution Strategies’’ (CMA-ES) algorithm [23]. This powerful evolutionary based optimizer is an improvement of the original $(\mu/\rho^+ \lambda)$ -ES variants [25], thanks to the covariance matrix adaptation property. Indeed, the CMA-ES allows generations of new individuals that are possibly not distributed along the principal axis of the covariance matrix. In the present work, we apply this evolutionary algorithm by using the recommended parameters values (step-size $\sigma = 0.5$, population size $\lambda = 4 + \lfloor 3 \log d \rfloor$, number of parents for recombination $\mu = \frac{\lambda}{2}$, etc.) of [23]. A complete tutorial of the CMA-ES algorithm, including a source code and best uses, is available in [26]. More details on GP meta-models can also be found in [24].

Let $\hat{\mathbf{x}}_n$ be the optimum of $\hat{f}(\mathbf{x})$. It is expected that $\hat{\mathbf{x}}_n \approx \mathbf{x}_{\text{opt}}$ if the approximation error $\hat{f} - f$ is small enough. The advantage of minimizing \hat{f} instead of f is that GP models are usually inexpensive to evaluate compared to the original objective function. To control the error in the approximation, one proceeds sequentially by adding progressively new points in the area of interest. A deterministic optimization procedure would choose the next point \mathbf{x}_{n+1} as the optimal point of \hat{f} . However, the GP model provides probabilistic information that can be exploited to propose more robust strategies based on merit functions, which combine the prediction and its variance. In this work, we use the Augmented Expected Improvement (AEI) merit function [27], which estimates the expected progress in the cost, taking into account the noise in the observed values and the prediction variance:

$$AEI(\mathbf{x}) = EI(\mathbf{x}) \left(1 - \frac{\sigma_\epsilon}{\sqrt{\hat{\sigma}_f^2(\mathbf{x}) + \sigma_\epsilon^2}}\right), \quad (6)$$

where the Expected Improvement $EI(\mathbf{x})$ is defined by

$$EI(\mathbf{x}) = \hat{\sigma}_f(\mathbf{x}) [u(\mathbf{x})\Phi(u(\mathbf{x})) + \phi(u(\mathbf{x}))], \quad (7)$$

$$u(\mathbf{x}) = \frac{\hat{f}(\mathbf{x}_{*,n}) - \hat{f}(\mathbf{x})}{\hat{\sigma}_f(\mathbf{x})}, \quad (8)$$

with Φ and ϕ the cumulative and density functions of the standard Gaussian distribution, and $\mathbf{x}_{*,n} \in \mathbf{X}_n$ is the current effective best solution (see [27]). The optimum \mathbf{x}_{n+1} of the AEI is added to \mathbf{X}_n , and f is evaluated at the new point providing y_{n+1} . Setting $n \leftarrow n + 1$, a new iteration can start updating the GP model. Overall, each iteration requires one computation of the cost and the resolution of two optimization problems: a first one for the hyper-parameters of the GP model, and a second one to find the AEI optimum. The iterations of the GP-based optimization problem are continued until a stopping criterion is satisfied or the resources allocated to the optimization have been exhausted. Obviously, if f is not computable for the proposed \mathbf{x}_{n+1} , the optimization process is stopped prematurely, because \mathbf{x}_{n+1} and y_{n+1} cannot be added to \mathbf{X}_n and \mathbf{Y}_n .

2.2. Classification method

As underlined in the introduction, the proposed approach based on a classification method is closely related to the handling of inequality constraints in the optimization problem (1). Indeed, the non-computable simulations are usually located in some specific regions of the design space. Therefore, the presence of non-computable points can be managed as inequality constraints if these regions of failures can be identified thanks to a probabilistic formulation. In this framework, a failure probability would be used within the optimization process to avoid the non-computable simulations.

EGO methods with inequality constraints were considered by [28]. The key idea followed in this reference is to rely on m additional surrogates to estimate the constraints $Q_i(\mathbf{x})$. For Gaussian Process models, one can easily determine the probability $P_i(Q_i \geq 0|\mathbf{x})$ that the constraint Q_i is satisfied at \mathbf{x} . Assuming the independence of the constraints probability, the consolidated probability $P(\mathbf{Q} \geq \mathbf{0}|\mathbf{x}) = \prod_{i=1}^m P_i(Q_i \geq 0|\mathbf{x})$ is obtained. This probability is used to modify the unconstrained AEI criterion and favor regions feasible with a probability:

$$AEI_{\mathbf{Q}}(\mathbf{x}) = AEI(\mathbf{x})P(\mathbf{Q} \geq \mathbf{0}|\mathbf{x}). \quad (9)$$

Although effective in many problems, this GP modeling of the constraints faces several limits. First, its computational cost increases with the number m of constraints and can be an issue for problems with large m . Second, the approximation by GP models assumes a sufficient smoothness of the Q_i . This rules out the case of binary constraints (feasible/infeasible). Another situation, non-amenable to the GP approximations of the constraints, is when the search domain Ω possesses a sub-domain Ω_{χ} over which f is non-computable, *i.e.* undefined, but there is no explicit expression or a priori knowledge of this sub-domain. As a result, Ω_{χ} cannot be expressed in terms of inequality constraints.

Classification methods recently proposed by [21] are better suited to deal with discontinuous and binary constraints in GP-based optimization procedures. Therefore, we aim at using this approach to define a unique framework to deal with inequality constraints and non-explicit feasible domain definition in presence of a non-computable domain $\Omega_{\chi} \subset \Omega$. In essence, classification is based on the reformulation of problem (1) into the unconstrained form

$$\min_{\mathbf{x} \in \Omega_{\text{adm}}} f(\mathbf{x}), \quad \Omega_{\text{adm}} := \Omega_{\mathbf{Q}} \setminus (\Omega_{\chi} \cap \Omega_{\mathbf{Q}}), \quad (10)$$

where $\Omega_{\mathbf{Q}} := \{\mathbf{x} \in \Omega, Q_{i=1,\dots,m}(\mathbf{x}) \geq 0\}$ is the subset of Ω satisfying the inequality constraints. As the admissible set Ω_{adm} has no explicit form, in general, we rely on a binary classifier with two classes C^+ and C^- over Ω , corresponding to the admissible (\mathbf{x} satisfies all constraints and $f(\mathbf{x})$ is computable) and non-admissible (\mathbf{x} does not satisfy all constraints or $f(\mathbf{x})$ is not computable) domains respectively. To construct this classifier, each \mathbf{x}_i of \mathbf{X}_n is equipped with a value $z_i = \pm 1$ depending on its membership C^{\pm} . To predict the class of a new point \mathbf{x} we introduce a classification function $h : \mathbf{x} \in \Omega \rightarrow \mathbb{R}$, such that $z(\mathbf{x}) = \text{sign}h(\mathbf{x})$. A Least-Squares Support Vector Machine (LS-SVM) [20] is used to construct h . The LS-SVM method extends the original Support Vector Machine (SVM) classifier [3] to quadratic penalization, resulting in a linear (but non-sparse) system to be solved (see below). Further details and discussion on SVM and LS-SVM methods can be found in [29].

The LS-SVM method [20] is a linear classifier, for C^+ and C^- , in a feature space induced by the transformation $\phi : \Omega \rightarrow \mathcal{F}$:

$$h(\mathbf{x}) = \mathbf{w}^T \phi(\mathbf{x}) + b. \quad (11)$$

Here, \mathbf{w} and $\phi(\mathbf{x})$ are the weights and features vectors, while $b \in \mathbb{R}$ is a constant. The feature space \mathcal{F} is generated by a reproducing kernel $r : \Omega \times \Omega \rightarrow \mathbb{R}$ representing the inner product between images in \mathcal{F} of vectors: $r(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x}) \cdot \phi(\mathbf{x}')$. We use the classical Gaussian kernel,

$$r(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\lambda^2}\right), \quad (12)$$

with scale factor $\lambda \in \mathbb{R}$ to be adjusted. The LS-SVM parameters (\mathbf{w}, b) satisfy the primal constrained optimization problem

$$\begin{aligned} \min_{\mathbf{w}, b, \mathbf{e}} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + \gamma \frac{1}{2} \sum_{i=1}^n e_i^2, \\ \text{s.t.} \quad & z_i = \mathbf{w}^T \phi(\mathbf{x}_i) + b + e_i, \quad i = 1, \dots, n, \end{aligned} \quad (13)$$

with trade-off parameter $\gamma \in \mathbb{R}^+$ (to be fixed) and relaxation variables e_i allowing for miss-classification. The Lagrangian of this optimal problem is

$$\mathcal{L}(\mathbf{w}, b, \boldsymbol{\alpha}_n, \mathbf{e}) = \frac{1}{2} \|\mathbf{w}\|^2 + \gamma \frac{1}{2} \sum_{i=1}^n e_i^2 - \sum_{i=1}^n \alpha_i (\mathbf{w}^T \phi(\mathbf{x}_i) + b + e_i - z_i), \quad (14)$$

where the $\alpha_i \in \mathbb{R}$ are the Lagrange multipliers of the constraints. Denoting $\mathbf{Z}_n = (z_1 \cdots z_n)^T$, the optimality conditions of the Lagrangian (15)

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \mathbf{0} & \longrightarrow \mathbf{w} = \sum_{i=1}^n \alpha_i \phi(\mathbf{x}_i), \\ \frac{\partial \mathcal{L}}{\partial b} = 0 & \longrightarrow \sum_{i=1}^n \alpha_i = 0, \\ \frac{\partial \mathcal{L}}{\partial \boldsymbol{\alpha}_n} = \mathbf{0} & \longrightarrow \alpha_i = \gamma e_i, \quad i = 1, \dots, n, \\ \frac{\partial \mathcal{L}}{\partial \mathbf{e}} = \mathbf{0} & \longrightarrow \mathbf{w}^T \phi(\mathbf{x}_i) + b + e_i - z_i = 0, \quad i = 1, \dots, n, \end{cases} \quad (15)$$

are used to derive a linear system for the dual model parameters b and $\boldsymbol{\alpha}_n = (\alpha_1 \cdots \alpha_n)^T$,

$$\begin{bmatrix} \mathbf{R} + \gamma^{-1} \mathbf{I} & \mathbf{1} \\ \mathbf{1}^T & 0 \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha}_n \\ b \end{bmatrix} = \begin{bmatrix} \mathbf{Z}_n \\ 0 \end{bmatrix}, \quad (16)$$

where $\mathbf{R} \in \mathbb{R}^{n \times n}$ is the kernel matrix with $R_{i,j} = r(\mathbf{x}_i, \mathbf{x}_j)$. This system is solved *via* a Cholesky factorization [29], and h in (11) is expressed in terms of dual model parameters to obtain

$$h(\mathbf{x}) = \sum_{i=1}^n \alpha_i r(\mathbf{x}_i, \mathbf{x}) + b. \quad (17)$$

The LS-SVM classifier depends on two parameters, γ and λ , to be determined when solving (16). This step is called *model selection* and it is usually performed with cross-validation approaches [30]. The most common method is the k -fold cross-validation, in which the training set is split into k disjoint subsets. Then, the LS-SVM model is sequentially trained on $k - 1$ subsets and the remaining subset is used to evaluate the classification efficiency by computing the error rate. The global k -fold estimate of the error rate is the average

of the k error rates. The extreme case where $k = n$ is called the Leave-One-Out (LOO) cross-validation and it allows an almost unbiased estimate of the error rate [31]. Unfortunately, the LOO computation is too expensive to be applied on common training sets. However, in the specific case of LS-SVM classifiers, it was shown [32, 33, 34] that exact LOO can be performed in only $\mathcal{O}(n)$ operations. In this work, we rely on this efficient LOO procedure to determine γ and λ that minimize the Predicted Residual Sum-of-Squares (PRESS) criterion [35],

$$\text{PRESS} = \sum_{i=1}^n \left(z_i - \widehat{z}_i^{(-i)} \right)^2, \quad (18)$$

where $\widehat{z}_i^{(-i)}$ is the predicted output $z(\mathbf{x}_i)$ of the LS-SVM, when the i -th training point (\mathbf{x}_i, z_i) is disregarded from the construction of the classifier. A simple expression for the predicted residuals is proposed in [34],

$$z_i - \widehat{z}_i^{(-i)} = \frac{\alpha_i}{D_{ii}^{-1}}, \quad i = 1, \dots, n, \quad (19)$$

where the D_{ii}^{-1} are the diagonal entries of the inverse of \mathbf{D} , the matrix of system (16). An explicit expression is also provided in [34] to compute D_{ii}^{-1} during the factorization of \mathbf{D} . The minimization of the PRESS (18) for λ and γ is performed with the CMA-ES algorithm [23] in all the examples presented below.

The LS-SVM binary classifier is finally extended to a probabilistic classification, relating h in (17) to the probability of the class C^+ , denoted $P(C^+|\mathbf{x})$. A comparison of several probability models for the LS-SVM classification is provided in [36]. We use the sigmoid function [37] and expressing the probability of C^+ as

$$P(C^+|\mathbf{x}) = \frac{1}{1 + \exp(Ah(\mathbf{x}) + B)}. \quad (20)$$

The parameters A and B of the sigmoid are determined by minimizing the probability of misclassification, see [37, 38]. In practice, the probability P goes to 1 (resp. 0) as the classifier is certain that $\mathbf{x} \in \Omega_{\text{adm}}$ and belongs to C^+ ($\mathbf{x} \notin \Omega_{\text{adm}}$ and belongs to C^-), while a value of $P = 1/2$ denotes a complete uncertainty in the classification. This can occur when \mathbf{x} is far from any observations in \mathbf{X}_n or close to the interface between the two classes.

2.3. EGO-LS-SVM method

2.3.1. Extended merit functions

We now return to the optimization problem and introduce the EGO-based method that we call EGO-LS-SVM for it is based on the LS-SVM classifier. The EGO-LS-SVM method uses the probability $P(C^+|\mathbf{x})$ to derive a merit function from the AEI, in order to select a new point $\mathbf{x}_{n+1} \in \Omega$. Following [28, 21], the selected point should present high expected improvement (relatively to another point) of the objective function value and a high probability of belonging to Ω_{adm} . These considerations lead us to propose the following sequential infilling strategy, selecting alternatively one of the following definitions for \mathbf{x}_{n+1} in the EGO-LS-SVM method:

$$\mathbf{x}_{n+1} = \arg \max_{\mathbf{x} \in \Omega} AEI(\mathbf{x})P(C^+|\mathbf{x}), \quad (21)$$

$$\mathbf{x}_{n+1} = \arg \max_{\mathbf{x} \in \Omega} AEI(\mathbf{x}) \quad \text{s.t.} \quad P(C^+|\mathbf{x}) \geq \rho, \quad (22)$$

$$\mathbf{x}_{n+1} = \arg \min_{\mathbf{x} \in \Omega} \widehat{f}(\mathbf{x}) \quad \text{s.t.} \quad P(C^+|\mathbf{x}) \geq \rho, \quad (23)$$

$$\mathbf{x}_{n+1} = \arg \max_{\mathbf{x} \in \Omega} [AEI(\mathbf{x})P(C^+|\mathbf{x}) (1 - P(C^+|\mathbf{x}))]. \quad (24)$$

The first definition in (21) corresponds to the extension of the AEI favoring points with high chance of feasibility. The second expression (22) maximizes the original AEI definition by enforcing a minimal probability ρ of feasibility; we use $\rho = 0.5$ in all subsequent computations. In (23), the predicted objective function is

directly minimized over the approximated feasible domain defined by $\{\mathbf{x} \in \Omega, P(C^+|\mathbf{x}) > \rho\}$. Finally (24) combines the two classes probabilities to favor areas where the classification is the most uncertain ($P \sim 0.5$), to improve the exploration along the estimated feasible domain boundaries.

Finally the new points \mathbf{x}_{n+1} are determined using CMA-ES algorithms without [23] or with constraints [39], depending on the considered definition. Whence the new point has been selected, the constraints Q_i and the objective function f are evaluated at the selected point. If $\mathbf{x}_{n+1} \notin \Omega_{\chi}$, that is $f(\mathbf{x}_{n+1})$ is computable we add $f(\mathbf{x}_{n+1})$ to the training set and update the GP model of f . Otherwise the new point is simply considered as a missing data and disregarded in the subsequent constructions of the GP model of f . In any cases, we set $z_{n+1} = +1$ if $f(\mathbf{x}_{n+1})$ is computable and $\mathbf{Q}(\mathbf{x}_{n+1}) \leq 0$, and $z_{n+1} = -1$ otherwise. Therefore, the GP model of f and the LS-SVM classification may involve different numbers of observations if some of them fall in Ω_{χ} . Note that if $\mathbf{x}_{n+1} \in \Omega_{\chi}$, we have $\hat{f}_{n+1} = \hat{f}$, but because of the update of $P(C^+|\mathbf{x})$ we shall have in general $\mathbf{x}_{n+2} \neq \mathbf{x}_{n+1}$. Consequently, the EGO-LS-SVM method will improve the estimation of Ω_{adm} till it proposes a new point $\mathbf{x}_{n+1} \in \Omega_{\text{adm}}$ to update \hat{f} .

Compare to the similar classification based approach proposed in [21], we remark that our approach is based on the original sigmoid probability in (34) such that $P(C^+|\mathbf{x})$ remains differentiable, facilitating the search of \mathbf{x}_{n+1} . Second, a single point is added and have to be evaluated per EGO iteration, where the approach in [21] enriches the training set with $d + 1$ new points at each iteration. The computational cost to perform these $d + 1$ evaluations of f is expected to be too important in many application and is not necessary for the LS-SVM classification procedure. This is demonstrated on a numerical example later, in section 3.5, where we discuss further the differences and respective efficiencies of the method of [21] and our.

The work-flow of one iteration step of the EGO-LS-SVM method is summarized in the Algorithm 1 below. In the full optimization process, the Algorithm 1 is called iteratively until a convergence is reached or a maximum workload has been reached. The lines 1 and 2 correspond to the GP surrogate construction of the objective function f . This allows computing the mean and variance predictions given in (4) and (5). Lines 3 to 5 concern the LS-SVM classification model with the classification probability in (20) as output. To do this, a first optimization problem is solved for the parameters (γ, λ) (line 3). Then, the system (16) is solved, at line 4, to determine (α_n, b) in (17). Finally, the coefficients A and B in (20) are computed by Newton's method [38]. To the end, the new point \mathbf{x}_{n+1} is determined at line 6 by solving a last optimization problem for one of the merit functions in (21-24).

Algorithm 1: Work-flow of one EGO-LS-SVM iteration.

Input: Sample set points \mathbf{X}_n , objective function values \mathbf{Y}_n , class labels \mathbf{Z}_n
Output: New selected point \mathbf{x}_{n+1}

!Build GP surrogate!
1 Find $(\Theta, \sigma_\epsilon^2)$ maximizing the *log-likelihood* !Using CMA-ES
2 Factorize $(\mathbf{C}(\Theta) + \sigma_\epsilon^2 \mathbf{I})$!With LU decomposition

!Build LS-SVM model!
3 Find (γ, λ) minimizing the *PRESS* criterion (18) !Using (19) L00 expression and CMA-ES
4 Compute (α_n, b) solving system (16) !Using Cholesky factorization
5 Find (A, B) minimizing the *misclassification probability* !Using Newton's method

!Merit function!
6 Find \mathbf{x}_{n+1} the *extended merit optimum* !Using one of equations (21-24) and CMA-ES

2.3.2. Convergence assessment

In the following sections, we are assessing and comparing the efficiency of the proposed EGO-LS-SVM method on several problems. To this end, we need to monitor the convergence of the sequence of optimization iteration. The convergence can be characterized in different ways. For instance one may report the best point

in the training set, $\mathbf{x}_{\text{best}} \in \mathbf{X}_n$, corresponding to the smallest objective function value $f_{\text{best}} \doteq f(\mathbf{x}_{\text{best}})$. Note that f_{best} is an upper bound of the true optimal value, and that f_{best} can only improve as the optimization iterations proceeds. Alternatively, in the analytical tests presented in the next section, where the objective function is easily evaluated, one can determine at each iteration the approximated optimal point $\mathbf{x}_{\text{opt}}^*$ and objective function value $f_{\text{opt}}^* \doteq f(\mathbf{x}_{\text{opt}}^*)$ associated to the current state of knowledge of f and computable domain $P(C^+|\mathbf{x})$. Specifically, one can solve for $\mathbf{x}_{\text{opt}}^*$ the following constrained optimization problem

$$\mathbf{x}_{\text{opt}}^* = \arg \min_{\mathbf{x} \in \Omega} \widehat{f}(\mathbf{x}) \quad \text{s.t.} \quad P(C^+|\mathbf{x}) \geq 0.5, \quad (25)$$

and subsequently evaluate $f_{\text{opt}}^* = f(\mathbf{x}_{\text{opt}}^*)$. It is noted, however, that $\mathbf{x}_{\text{opt}}^*$ may not be a computable point since the classifier may have a non-zero misclassification rate. Further, when the exact solution of the optimization problem is known, one can report the distance to current estimate $\mathbf{x}_{\text{opt}}^*$ to the optimum true optimum \mathbf{x}_{opt} , in terms of Euclidian distance $\|\mathbf{x}_{\text{opt}} - \mathbf{x}_{\text{opt}}^*\|$ or objective function values $f(\mathbf{x}_{\text{opt}}^*) - f(\mathbf{x}_{\text{opt}}) \geq 0$.

2.3.3. Illustration of the infilling strategy

Before applying the EGO-LS-SVM method to generic problems, we first illustrate the properties of the different merit functions in the criteria (21-24), and the benefit of applying them in a sequential fashion. To this end, we consider the one-dimensional domain $\Omega = [0,1]$ and the *Ricker Wavelet* objective function $f(x) = (1 - 2\pi^2 x^2) \exp(-\pi^2 x^2)$. We define the set of non-computable points through

$$\Omega_{\chi} = \{x \in \Omega : |x - x_{\min}| < 0.2\}, \quad (26)$$

where $x_{\min} = \frac{\sqrt{3/2}}{\pi}$ is the minimum of f in Ω . Letting $\Omega_{\text{adm}} = \Omega \setminus \Omega_{\chi}$, the optimization problem becomes

$$\min_{x \in \Omega_{\text{adm}}} f(x) = (1 - 2\pi^2 x^2) \exp(-\pi^2 x^2). \quad (27)$$

The optimum of (27) is found on the boundary of Ω_{adm} at $x_{\text{opt}} = x_{\min} + 0.2$. We solve this problem for several infilling strategies, with an initial sample set with size $n_{\text{LHS}} = 10$ observations, and fix the maximum number of calls to f to $N_f = 50$ (thus 40 optimization iterations are performed). Figure 1 compares the distances to the optimum as a function of N_f for the sequential use of all the merit functions (21-24), or using systematically only one of them. It is seen that using only (21) (pink curve) or (24) (black curve) to drive the procedure does not yield satisfactory results on this example. In contrast, using (22) alone (blue curve) or (23) alone (green curve) yields a convergence similar to that obtained using the four merit functions sequentially (red curve). These behaviors can be explained on this simple problem: because the criteria (22) and (23) produce points closer to the boundary than the unconstrained merit functions in (21) and (24), they are more efficient to recover an optimum located on the boundary. Nevertheless, using sequentially all the merit functions (21-24) does not impact negatively the convergence. We thus advise for the alternative use of all the four merit functions in a sequential infilling strategy. This setting favors both the exploration of promising areas (with large AEI) of the design space and of the most uncertain boundaries of the computable domain ($P \sim 0.5$).

Finally, we report in Figure 2 the convergence of EGO-LS-SVM when using the merit functions (22) alone (left plot), or (23) alone (right plot), and for different values of the threshold probability $\rho = 0.2, 0.5$ and 0.8 . Again, the curves depict the distance to the optimum as a function of the total number N_f of calls to f . These plots show that the value of ρ seems to have a moderate effect on the convergence in the present example. In addition, using $\rho = 0.5$ appears to be the best choice for the two merit functions. This value will be considered as the default value in the following experiments.

3. Analytical test problems

In this section, we illustrate the accuracy and the computational efficiency of the EGO-LS-SVM approach by solving few optimization problems defined analytically. We consider problems with objective functions

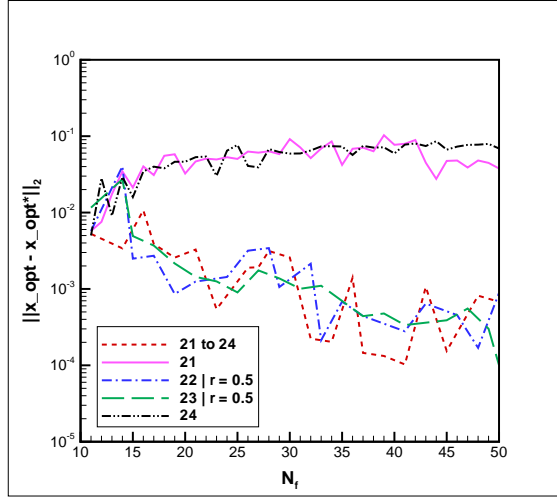
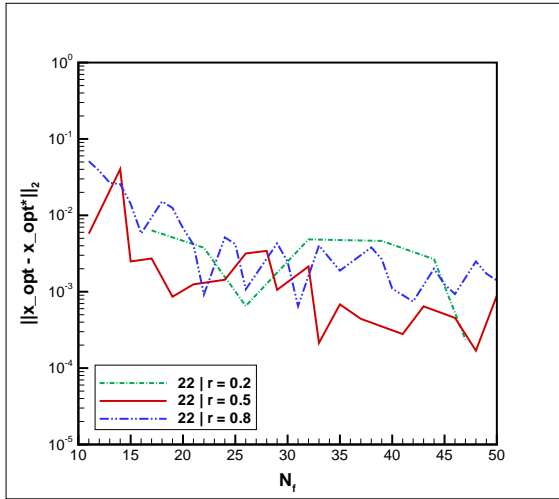
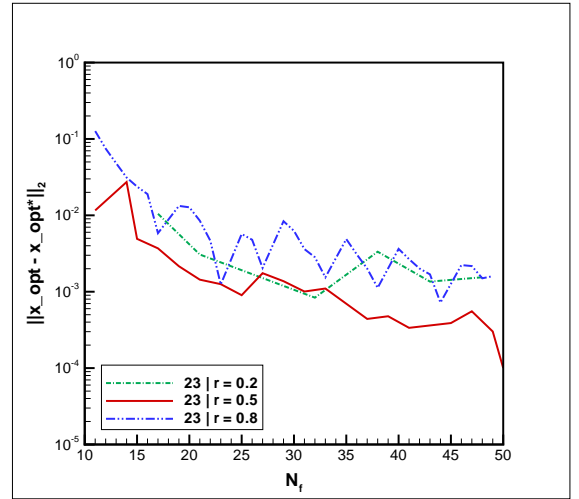


Figure 1: Convergence with the number of calls N_f of the distance to optimum, $\|\mathbf{x}_{\text{opt}} - \mathbf{x}_{\text{opt}}^*\|$. Values are only reported for computable optimum $\mathbf{x}_{\text{opt}}^*$ solution of (25). Case of problem (27).



(a) Merit function (22).



(b) Merit function (23).

Figure 2: Convergence with the number of calls N_f of the distance to optimum, $\|\mathbf{x}_{\text{opt}} - \mathbf{x}_{\text{opt}}^*\|$, for the infilling strategy based on single merit function (22) (left) or (23) (right), and for different threshold probabilities ρ . Values are only reported for computable optimum $\mathbf{x}_{\text{opt}}^*$ solution of (25). Case of problem (27).

that are not computable over a subset Ω_{χ} of the full design space Ω . The classification approach is used to estimate the admissible (*i.e.* computable) domain $\Omega_{\text{adm}} := \Omega \setminus \Omega_{\chi}$, and the GP model of f is constructed using only evaluations at points proposed in Ω_{adm} . For reference and comparison purposes, we rely on the evolutionary algorithm CMA-ES [23] to solve the different problems, employing a high penalization value $f(\mathbf{x}) = f_{\infty}$ for the individual $\mathbf{x} \notin \Omega_{\text{adm}}$. This penalization f_{∞} and other parameters of the CMA-ES method (population size, cross-over parameter...) are carefully set for each optimization problem, following the recommendations [23].

The selected test problems are presented in the following. A simple two-dimensional illustrative example is firstly proposed in Section 3.1; a multi-dimensional optimization problem is considered in Section 3.2,

which includes optimizations in increasing dimensions and calculations with different initial sample set sizes; Section 3.3 investigate the behavior of the EGO-LS-SVM method in the case of a multi-dimensional optimization problem with complex domain Ω_χ corresponding to a large portion of Ω (case of an over-constrained problem); Section 3.4 proposes the minimization of a problem where the domain Ω_{adm} and objective function f are defined probabilistically; finally Section 3.5 concerns the efficiency comparison of the original SVM based approach [21] to our new EGO-LS-SVM method.

3.1. Two-dimensional illustrative example

We consider the two-dimensional domain $\Omega = [0,4]^2$ and the simple convex objective function $f(\mathbf{x}) = x_1^2 + x_2^2$. We define the set of non-computable points through

$$\Omega_\chi = \{\mathbf{x} \in \Omega : c_1(\mathbf{x}) < 0, c_2(\mathbf{x}) < 0\}, \quad (28)$$

where $c_1(\mathbf{x}) = 0.25x_1^2 + 0.75x_2^2 - 1$ and $c_2(\mathbf{x}) = 0.75x_1^2 + 0.25x_2^2 - 1$. Letting $\Omega_{\text{adm}} = \Omega \setminus \Omega_\chi$, the optimization problem becomes

$$\min_{\mathbf{x} \in \Omega_{\text{adm}}} f(\mathbf{x}) = x_1^2 + x_2^2. \quad (29)$$

The exact optimum of the problem is found to be on the boundary of Ω_{adm} at $\mathbf{x}_{\text{opt}} = (1,1)$.

The EGO-LS-SVM algorithm is initialized using a Latin Hypercube Sampling (LHS) set of 15 points drawn randomly in Ω . The optimization iterations are repeated till the AEI merit function reaches a value less than 10^{-3} . In the CMA-ES method, the optimization is carried-out until the estimated optimum is in Ω_{adm} with a value of f less than the EGO-LS-SVM solution. As mentioned before, a large penalization $f_\infty = 100$ is applied to points not belonging to Ω_{adm} in the CMA-ES algorithm.

Figure 3 shows the convergence of the error (red squares and blue triangles, left axis) with the cumulated number of calls to (attempts to evaluate) f , denoted N_f . Note that N_f is also accounting for calls to f with $\mathbf{x} \notin \Omega_{\text{adm}}$. The error on the solution is defined as the Euclidian distance between the current estimate $\hat{\mathbf{x}}_{\text{opt}}$ of the optimum and the exact one, \mathbf{x}_{opt} . For the EGO-LS-SVM method, $\hat{\mathbf{x}}_{\text{opt}}$ is the solution of (29) where f is substituted with its GP model. For the CMA-ES approach, the distance to the exact optimum of each member of the population in Ω_{adm} is reported. The number N_{adm} of computable calls to f is also reported for both methods, using red and blue lines (right axis) in the same Figure. The EGO-LS-SVM (squares and red line) terminated at $N_f = 142$, with a number of computable calls equal to 91, such that roughly 50 points have been proposed in the non-feasible region Ω_χ . The evolutionary method needs 599 calls to f before it finds a better optimum than the EGO-LS-SVM one, among which 401 were for computable points. This result shows that, in this example, the EGO-LS-SVM method is able to properly estimate the feasible domain Ω_{adm} in the neighborhood of the optimum, and converges to this optimum at a significantly faster rate than the CMA-ES approach.

To better appreciate the effectiveness of the classification approach, contours of $P(C^+|\mathbf{x})$ are reported in Figure 4 at $N_f = 50$ and 100. Also shown in the plots are the iso-value corresponding to $P(C^+|\mathbf{x}) = 0.5$ (dashed white line), proposed computable points (white squares) and non-computable points (black triangles). After 50 calls to f , the classification has well discovered Ω_{adm} and the level $P(C^+|\mathbf{x}) = 0.5$ is not far from the actual boundary of Ω_{adm} . After 100 calls to f , a large fraction of the proposed points are concentrated around the optimum and the contour $P(C^+|\mathbf{x}) = 0.5$ is not following anymore the boundary of Ω_{adm} , except in the immediate neighbor of the optimum. Farther from the optimum in areas less explored, the classification probability quickly returns to $P(C^+|\mathbf{x}) \approx 0.5$ as we move away from observed points. This behavior is due to the classification parameters λ and γ (see Section 2.2) that are adjusted to capture the steep transition in $P(C^+|\mathbf{x})$ at the interface between Ω_{adm} and Ω_χ , where the optimum point \mathbf{x}_{opt} is located. This behavior is desirable because it allows considering the exploration of areas that were initially classified as unfeasible with high probability if they exhibit large merit function AEI. Indeed, the plot also reveals several calls to f for points in Ω_χ , far from the optimum, that confirm non-feasibility and balance large merit function values (recall that the unconstrained optimum is at the origin).

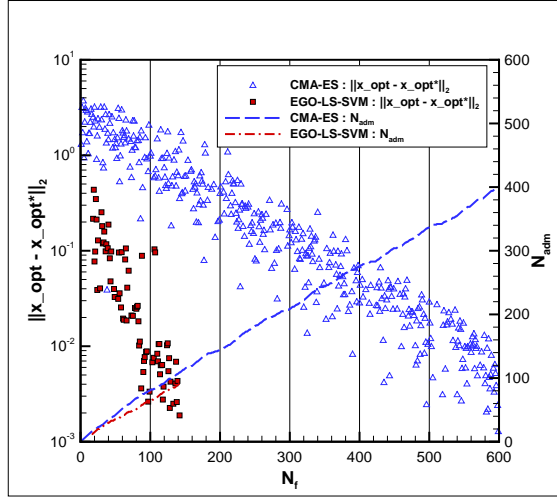


Figure 3: Distance to the optimum ($\|\mathbf{x}_{\text{opt}} - \mathbf{x}_{\text{opt}}^*\|$) and cumulated number of computable calls to f (N_{adm}) as functions of the total number of calls N_f and for the EGO-LS-SVM and CMA-ES optimization procedures. Distance is reported only for computable $\mathbf{x}_{\text{opt}}^*$ solution of (25). Case of problem (29).

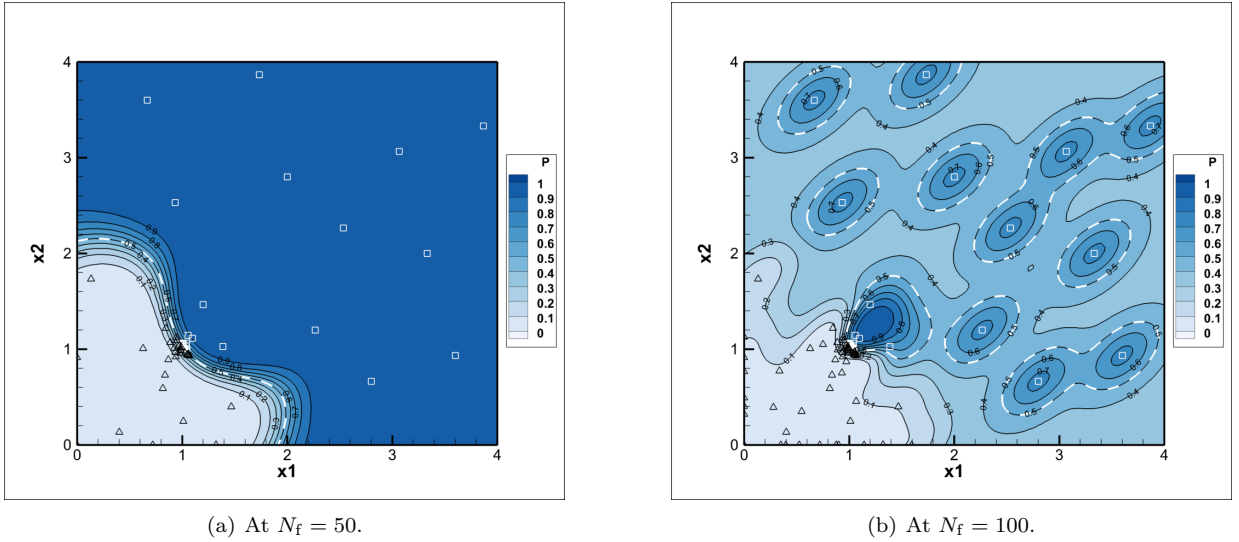


Figure 4: Contours of the classification probability $P(C^+|\mathbf{x})$ after 50 and 100 calls. Case of problem (29).

3.2. Higher dimensional problems

3.2.1. Influence of dimensionality

To investigate the behavior of the EGO-LS-SVM method as the dimensionality of the search space Ω is increasing, we propose the following problem where no function value is provided for points not belonging to the feasible domain. Let $d > 1$ be the dimensionality of $\Omega := [-1, 1]^d$. We denote B_d the hypersphere center on $\mathbf{x}_c = (0.1 \cdots 0.1)$ with radius $R_d = \sqrt{0.05(d-1)}$ and set $\Omega_\chi := B_d$. The optimization problem is then defined by

$$\min_{\mathbf{x} \in \Omega \setminus B_d} f(\mathbf{x}) = \sum_{i=1}^d x_i^2. \quad (30)$$

As for the previous problem, the optimal solution is found on the boundary of Ω_{χ} . The initialization of the EGO-LS-SVM method uses LHS set having increasing size with d . Specifically the tests shown below correspond to $d = 2, 5, 10$ and 20 and respective LHS sizes $n_{\text{LHS}} = 15, 40, 100$ and 200 . The stopping criteria for the EGO-LS-SVM and CMA-ES methods are identical to the previous one in Section 3.1.

Figure 5 shows the convergences for the different dimensionality d tested. The plot shows the sequence of values $f(\mathbf{x})$ at computable proposed points \mathbf{x} as a function of N_f . The evolutions of the cumulated number of computable calls N_{adm} are also reported.

For all the dimensionality d tested, the CMA-ES method needs more function calls than the EGO-LS-SVM method and the ratio of numbers of calls increases with d . In other words, the efficiency of the EGO-LS-SVM method relatively to the CMA-ES method is increasing with d . It is remarked that for the same number of calls to f , the CMA-ES is proposing more points in Ω_{adm} than the EGO-LS-SVM method, but the latter is more effective at proposing points with low f value. For the EGO-LS-SVM method, we also note the presence of distinct branches of function values. These branches reflect the competing trends involved in the selection of the new optimum candidates \mathbf{x}_{n+1} , because of the alternated use of conservative or aggressive strategies specified by (23) and (24).

3.2.2. Influence of initial sample set size

This section aims at assessing the impact of the initial sample set size on the convergence of the optimization procedure. Indeed, the initial design of experiments can be crucial in the optimization process and may significantly affect the EGO procedure. To study the robustness of the EGO-LS-SVM method, we solve the problem (30), with $d = 5$ and three different initial sample set sizes: $n_{\text{LHS}} = 10, 50$ and 100 . For these three cases, we perform 300 EGO iterations.

Figure 6 shows the convergences for the different sample set sizes n_{LHS} . The plot shows the sequence of objective function values f at computable points as a function of N_f (also counting the initial sample set generation). We see that for the $n_{\text{LHS}} = 50$ and 100 cases the algorithm immediately proposed new points that improve the function values of f and accordingly converge quickly within ≈ 200 iterations. The case of only $n_{\text{LHS}} = 10$ initial samples exhibits a very different behavior with an initial exploration phase with selected points having high objective function values, up to $N_f \approx 150$. after which the optimization quickly converges. Overall, the convergence requires slightly more iterations for $n_{\text{LHS}} = 10$ than for the larger initial sample sets. Nevertheless, this example highlights the robustness of the EGO-LS-SVM method since the procedure converges to the optimum even when a coarse initial sample set (10 points in 5 dimension) is employed.

3.3. Over-constrained problem

The following problem is designed to illustrate the limits of the EGO-LS-SVM method in the case where the feasible domain Ω_{adm} is hard to learn and explore, because of a complex structure. This problem which was proposed in [40] is then said over-constrained. The domain Ω_{adm} is here define as a subset of $\Omega = [-10,10]^d$, for $d = 7$, through

$$\Omega_{\text{adm}} = \{\mathbf{x} \in \Omega, c_{i=1,\dots,4}(\mathbf{x}) \leq 0\}, \quad (31)$$

where

$$\begin{aligned} c_1(\mathbf{x}) &= 2x_1^2 + 3x_2^4 + x_3 + 4x_4^2 + 5x_5 - 127, & c_2(\mathbf{x}) &= 7x_1 + 3x_2 + 10x_3^2 + x_4 - x_5 - 282, \\ c_3(\mathbf{x}) &= 23x_1 + x_2^2 + 6x_6^2 - 8x_7 - 196, & c_4(\mathbf{x}) &= 4x_1^2 + x_2^2 - 3x_1x_2 + 2x_3^2 + 5x_6 - 11x_7. \end{aligned}$$

The optimization problem finally writes as

$$\begin{aligned} \min_{\mathbf{x} \in \Omega_{\text{adm}}} f(\mathbf{x}) &= (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10x_5^6 \\ &\quad + 7x_6^2 + x_7^4 - 4x_6x_7 - 10x_6 - 8x_7. \end{aligned} \quad (32)$$

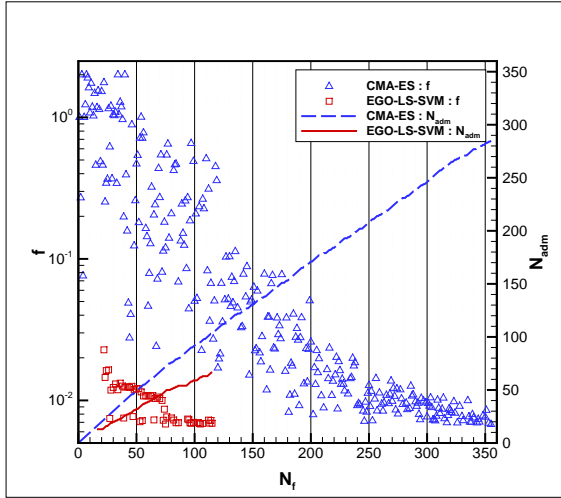
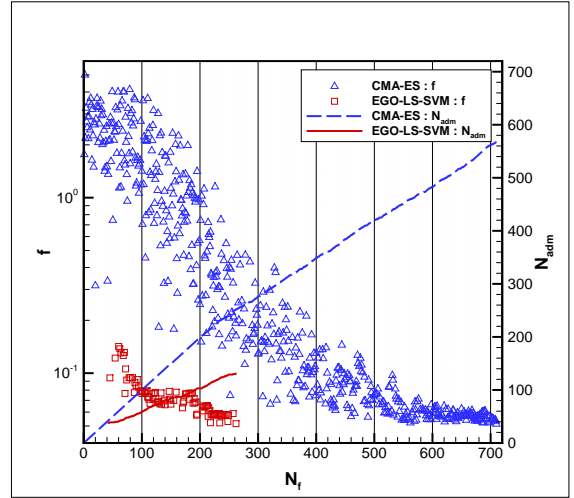
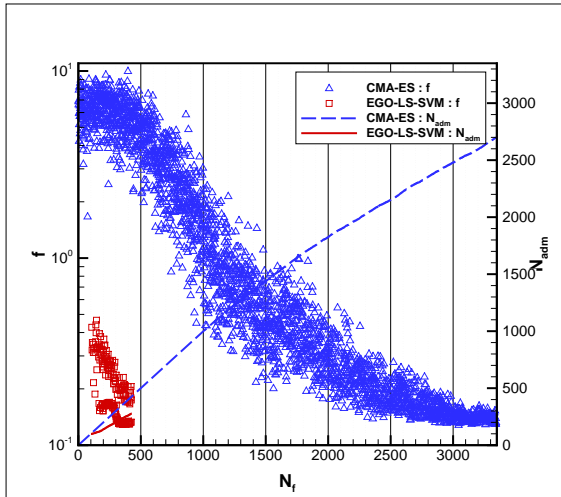
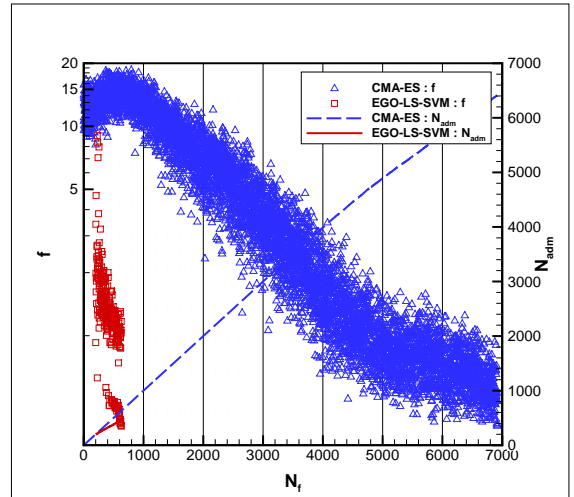
(a) $d = 2$.(b) $d = 5$.(c) $d = 10$.(d) $d = 20$.

Figure 5: Sequence of computable values of $f(\mathbf{x}_n)$ and cumulated number of computable calls N_{adm} as functions of N_f and for the EGO-LS-SVM and CMA-ES optimization procedures. Case of problem (30) with different dimensionality d as indicated.

The global optimum is located at

$$\mathbf{x}_{\text{opt}} = (2.330499, 1.951372, -0.4775414, 4.365726, -0.624487, 1.038131, 1.594227).$$

As before, only the proposed points in Ω_{adm} are evaluated and used to approximate f , while the classification proceeds with all the points, computable or not. Therefore, a large LHS set of 800 points drawn uniformly in Ω is needed to ensure that the initial GP model of f uses sufficiently many points. In the experiment presented below, only 7 of the 800 LHS points were drawn in Ω_{adm} , a low fraction illustrating the complexity of the constrained problem.

Figure 7 shows the evolution of the Euclidian distances to the exact optimum, $\|\mathbf{x}_{\text{opt}} - \hat{\mathbf{x}}_{\text{opt}}\|_2$ as a function of N_f . The EGO-LS-SVM method progresses during the first 500 calls to f (EGO iterations), but subsequently stalls. Note that when the optimization procedure is halted, only 187 evaluations have been generated for a total of 3541 calls to f . The low ratio of computable proposed points \mathbf{x}_n indicated that for

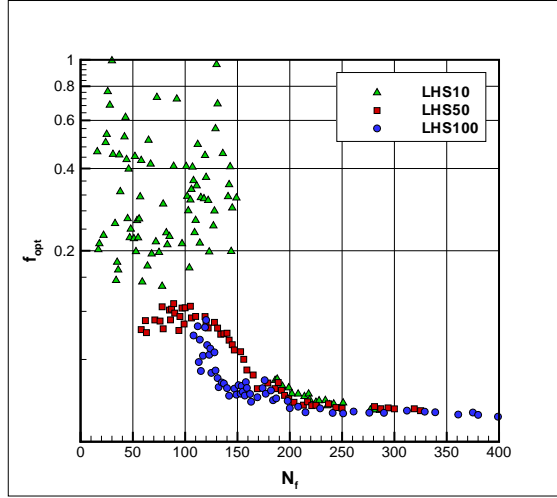


Figure 6: Sequences of optimum computable values of f_{opt} , as a function of N_f , for different initial sample set sizes n_{LHS} . Case of problem (30) with $d = 5$.

this example the EGO-LS-SVM method is not able to properly learn the domain Ω_{adm} . On the contrary, the CMA-ES method performs satisfactorily on this problem and appears to be much more robust against the complexity of Ω_{adm} . Nevertheless, it should be underlined that this problem corresponds to an extreme situation where the objective function is not computable over most of the search domain Ω .

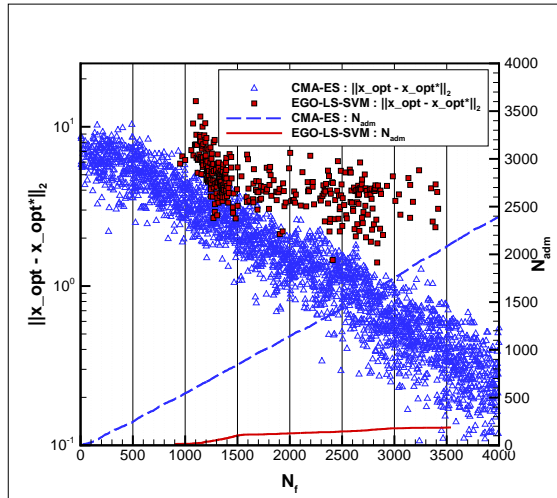


Figure 7: Distance to the optimum $\|\mathbf{x}_{\text{opt}} - \mathbf{x}_{\text{opt}}^*\|$ and cumulated number of computable calls to f N_{adm} as functions of N_f and for the EGO-LS-SVM and CMA-ES optimization procedures. Case of problem (32).

3.4. Problem with random computable domain

To complete the series of tests on analytically defined problems, we propose a problem for which the domain of non-computable function Ω_{χ} and the objective function are defined randomly. We set $\Omega = [-5, 5]^2$ and generate randomly Ω_{χ} using the following procedure. We first generate randomly 100 points $\mathbf{x}_{i=1, \dots, 100}^c \in \Omega$. Then, for each point \mathbf{x}_i^c we decide with a probability $p_{\chi} \in [0, 1]$ if f is non-computable in the neighborhood $\|\mathbf{x} - \mathbf{x}_i^c\|_2 \leq 1$. We then define Ω_{χ} as the intersection of Ω with the non-computable

neighborhoods. For this construction, the probability P_χ that f is not computable at a given $\mathbf{x} \in \Omega$ is approximated from the cumulative distribution function of the binomial law $B(100, \frac{\pi}{|\Omega|} p_\chi)$, where $|\Omega|$ is the volume of Ω . Figure 8 presents two random realizations of Ω_χ (grey areas) for a probability $P_\chi = 27\%$ and 61% in the left and right plot respectively.

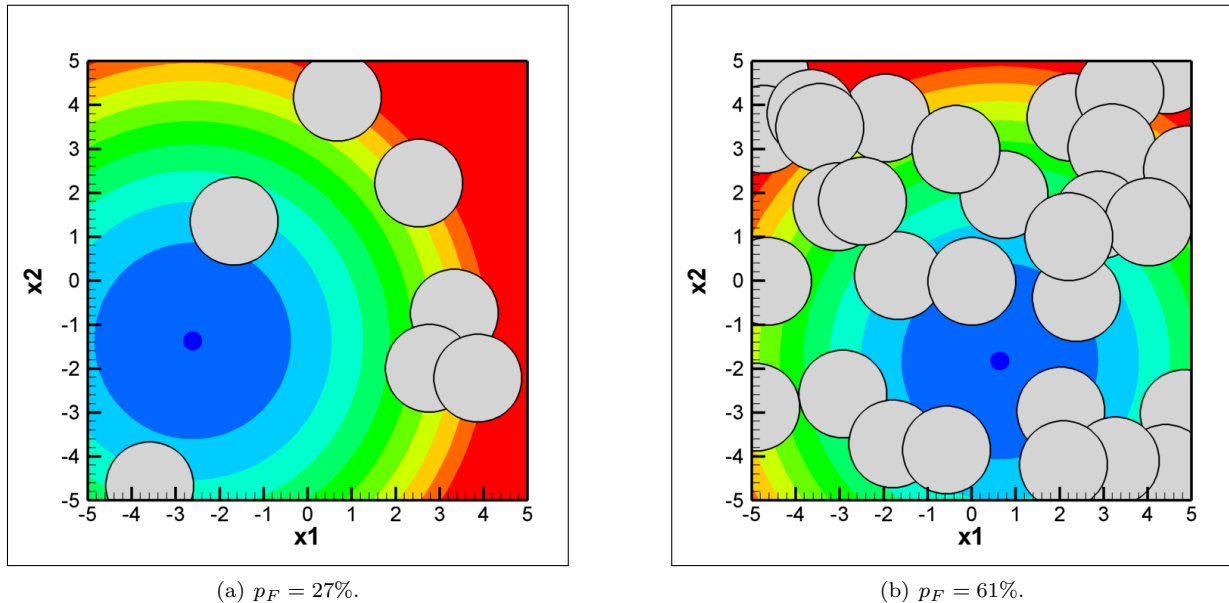


Figure 8: Realizations of the non-computable domain Ω_χ (grey circles) for probabilities $P_\chi = 0.21$ and 0.61 . Also shown are color contours of f .

The optimization problem is finally defined as

$$\min_{\mathbf{x} \in \Omega \setminus \Omega_\chi} f(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}_{\text{opt}}\|_2^2, \quad (33)$$

where the solution \mathbf{x}_{opt} is drawn at random in $\Omega \setminus \Omega_\chi$. In the computations, the EGO-LS-SVM method is initialized with an LHS set of $n_{\text{LHS}} = 10$ points in Ω and the optimization procedure is stopped at $N_f = 50$.

We solve the optimization problem (33) for several probabilities P_χ . For every value of P_χ several realizations of Ω_χ and \mathbf{x}_{opt} are generated to estimate statistics of the solution. Figure 9 shows the convergence of the optimization procedure with N_f (optimization iteration) and for increasing values of P_χ . Figures 9(a)–9(c) correspond to increasing values of P_χ and reports the median (blue line) and first and third quartiles (red lines) of the distance to the optimum. Figure 9(d) compares the convergence of the median distance to the optimum, for several values of P_χ . The plots show that as P_χ increases from 0, the convergence of the median and first quartile is only mildly affected. In contrast, the third quartile is significantly impacted, with an $O(1)$ distance to the optimum when $P_\chi = 61\%$. This behavior indicates that the EGO-LS-SVM method is quite robust and successful in finding the optimum, except when P_χ becomes too large with computable domains Ω_{adm} with a low ratio of $|\Omega_{\text{adm}}|/|\Omega|$ and complicated structures. In these situations, the EGO-LS-SVM method is not making progress and becomes inefficient at proposing new points $\in \Omega_{\text{adm}}$ to improve the model of f . As a result, the approach is either successful or completely failing, with a significantly skewed distribution of the distance to the optimum as evidenced by the distance between the first and third quartiles to the median. However, the results reported in Figure 9(d) show that the EGO-LS-SVM method is quite robust since for probabilities as high as $P_\chi = 79\%$ the approach is still converging for at least half of the generated problems.

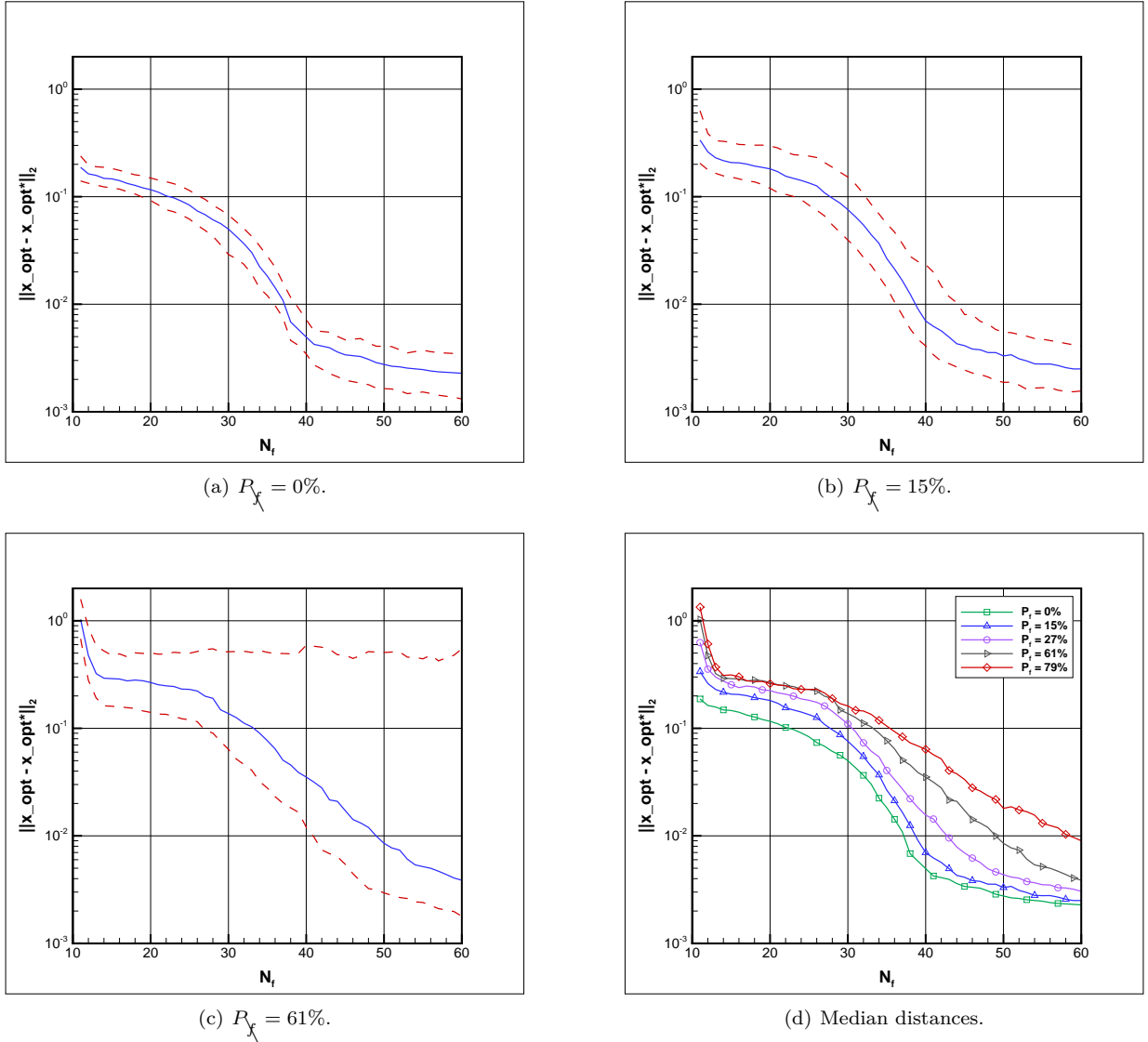


Figure 9: Statistic (median, 25% and 75% quartiles) of the distance to the optimum $\|\mathbf{x}_{\text{opt}} - \mathbf{x}_{\text{opt}}^*\|_2$ as a function of N_f and for several failure probabilities P_χ .

3.5. Comparison with EGO-SVM($d+1$)

A classification approach was proposed in [21] for the treatment of constraints, and it can be adapted to the case of non-computable domains. This section aims at contrasting this approach with ours, and at highlighting the improved efficiency of our approach.

As mentioned before, the method proposed in [21] uses a classifier to predict the feasibility of new points (*i.e.* whether constraints are satisfied), and so can be used in the same way to predict if f is computable. Compared to our approach, the method in [21] differs on four main points. First, it uses an SVM method to determine the classification function $h(\mathbf{x})$, in a polynomial space, where we use an LS-SVM method in general Reproducing Kernel Hilbert spaces. The LS-SVM has thus generally better approximation properties when the SVM method selects a subset of points to construct h with possibly a reduced computational cost. Second, using a reduced training set in SVM calls for a correction of the classification probability to ensure

that disregarded training samples are correctly classified. In [21] the following correction of the sigmoid function (20) was considered,

$$P(C^+|\mathbf{x}) = \frac{1}{1 + \exp\left(Ah(\mathbf{x}) + B\left(\frac{d^-(\mathbf{x})}{d^+(\mathbf{x})+\delta} - \frac{d^+(\mathbf{x})}{d^-(\mathbf{x})+\delta}\right)\right)}, \quad (34)$$

where $d^+(\mathbf{x})$ (resp. $d^-(\mathbf{x})$) is the distances of \mathbf{x} to the closest point in the training set belonging to C^+ (resp. C^-), and $\delta = 10^{-10}$ is used to avoid numerical overflows. This modification is introduced to account for the fact that the class of the training samples is exactly known. Third, the method in [21] considers a single merit function based on the expected improvement (EI). Forth and finally, the infilling strategy in [21] adds not only the point \mathbf{x}_{n+1} classified in C^+ and maximizing the EI but also d additional new points in its neighborhood to maintain the isotropy of the training set (a requirement made necessary by the corrected probability in (34)). In contrast, our approach uses the simpler and smoother sigmoid function, alternates between the criteria in (21-24) to accommodate different situations and enrich the training set by a single element at each iteration of the iterative procedure. As subtle as they may appear, these differences greatly impact the efficiency of the optimization procedures.

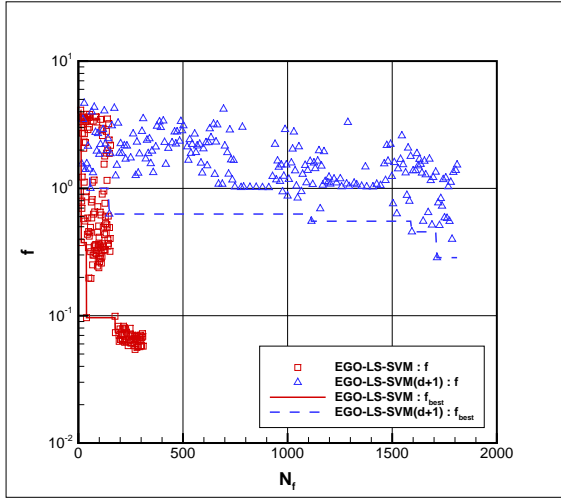
To distinguish between the impacts of the classifier construction methods and of the infilling strategies (classification probability, selection criteria for \mathbf{x}_n), we considered the following methods:

- EGO-LS-SVM : the approach proposed in this paper, with LS-SVM construction of the classification function $h(\mathbf{x})$, sigmoid classification probability definition, sequential selection using criteria (21-24).
- EGO-SVM($d+1$) : the approach proposed [21], with SVM construction of $h(\mathbf{x})$, modified sigmoid classification probability definition in (34), EI-only selection criteria and d -points neighborhood exploration.
- EGO-SVM : the same as EGO-LS-SVM but with a SVM construction of $h(\mathbf{x})$.
- EGO-LS-SVM($d+1$) : the same as EGO-SVM($d+1$) but with a LS-SVM construction of $h(\mathbf{x})$.

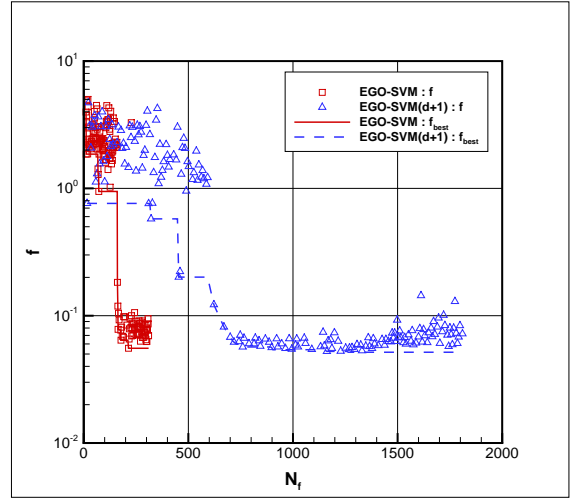
These methods are tested on problem (30), fixing $d = 5$. The results of these experiments are summarized in Figure 10. Note that the SVM and LS-SVM methods both use kernel-based constructions. The plots report the sequence of computable function values $f(\mathbf{x}_n)$ as a function of the number of function calls N_f . In the case of the EGO-SVM($d+1$) and EGO-LS-SVM($d+1$), the function values are reported only for the newly selected point \mathbf{x}_{n+1} , if computable, and not for its d additional companion points. In addition, the convergence of the best feasible function value f_{best} is reported, using lines, and considers all evaluations of f (including the eventual d additional points).

Figure 10(a) compares the LS-SVM methods for an initial sample set with $n_{\text{LHS}} = 10$ points. It is seen that our approach EGO-LS-SVM explores the design space during about 100 function calls and converges to the computable optimum in about 250 calls. In contrast, the EGO-LS-SVM($d+1$) method that uses the probability function and infilling strategy of [21] is seen to have not converged even after 1,800 function evaluations, being unable to focus on the correct area. Figure 10(b) shows the case of the SVM methods using again $n_{\text{LHS}} = 10$ samples. The plot shows that the EGO-SVM method converges essentially as fast as for our proposed EGO-LS-SVM, albeit with a somehow longer exploration phase, as one would have expected for using the SVM in place of the LS-SVM model. For the EGO-SVM($d+1$) method, corresponding to the original method proposed in [21], we observe that it is now converging too, but it requires a few times more function evaluations to achieve a comparable best value f_{best} compared to our infilling strategy. This result highlights the computational overhead due to the need of adding d additional points to maintain the isotropy of the sample points. Figure 10(c) (resp. 10(d)) corresponds to the same methods as in Figure 10(a) (resp. 10(b)) but now using an initial sample set with $n_{\text{LHS}} = 100$ points. It shows that our infilling method does not improve significantly when increasing n_{LHS} , or conversely demonstrating once more the robustness and interest of alternating between merit criteria.

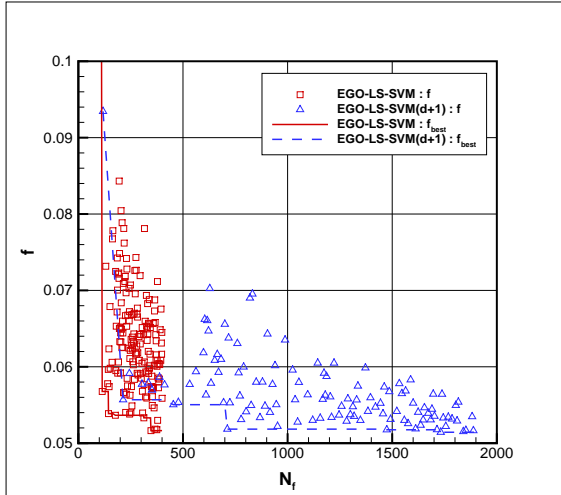
We finally remark that the construction of the SVM classifier is much more time consuming than the LS-SVM one, in the presented experiments using a kernel-based approach. This can be appreciated from the



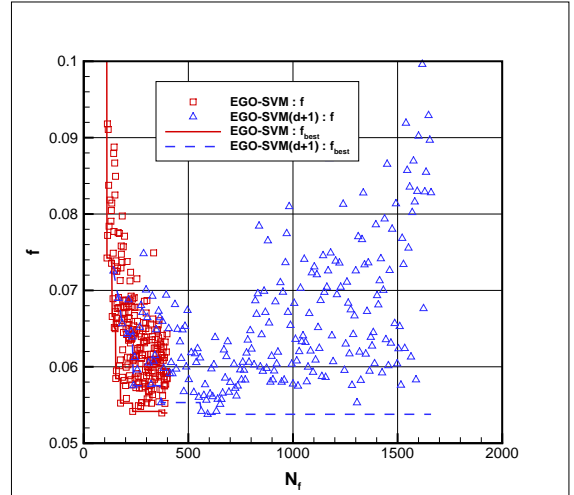
(a) LS-SVM construction, $n_{\text{LHS}} = 10$.



(b) SVM construction, $n_{\text{LHS}} = 10$.



(c) LS-SVM construction, $n_{\text{LHS}} = 100$.



(d) SVM construction, $n_{\text{LHS}} = 100$.

Figure 10: Sequence of computable objective function values $f(\mathbf{x}_n)$ (symbols) and convergence of the best value f_{best} (lines) as functions of the total number of calls N_f for different classifiers constructions, infilling strategies and initial sample set sizes. Case of problem (30) with $d = 5$.

computational times reported Table 1 for different training set sizes. The differences in the computational times are mainly due to *model selection* step in the SVM classifier, which involves the resolution of a quadratic problem (using an SMO method [41]). The higher computational cost of the SVM classifier construction further pleads for using the LS-SVM.

Training set size	50	100	150	200
CPU time (s) of LS-SVM	0.2	0.4	1.1	2.1
CPU time (s) of SVM	0.9	3.7	15.5	30.9

Table 1: Measured computational times (in seconds) for building the LS-SVM and SVM classifiers on training set with different sizes.

4. Application to flexible hydrofoil optimization

4.1. Optimization problem and solvers

We now apply the EGO-LS-SVM on a realistic problem corresponding to the optimization of a flexible hydrofoil. The objective of the optimization is to minimize the hydrofoil drag force at selected conditions (forward speed and lifting force) while ensuring non-cavitating flows. The complete description of the optimization problem and the hydrodynamical analysis of the optimized hydrofoil are not provided in this paper, which focuses on the method; interested readers can refer to [42]. The original optimization problem proposed in [42] involved 11 design variables describing the shape and elastic characteristics of the foil flexible trailing edge, as schematically illustrated in Figure 11. In the present work, the number of design variables is reduced to 5 in order to maintain reasonable computational times and to allow extensive comparisons of the optimization methods. Among the retained optimization variables, 4 concern the trailing edge camber geometry at rest (p_1, \dots, p_4), while the last one modifies the Young modulus E_{Beam} of the flexible part. We set $\Omega = [-0.1, 1.5] \times [-0.5, 0.2] \times [-0.3, 0.5] \times [-0.3, 0.3] \times [0.1, 1.2]$.

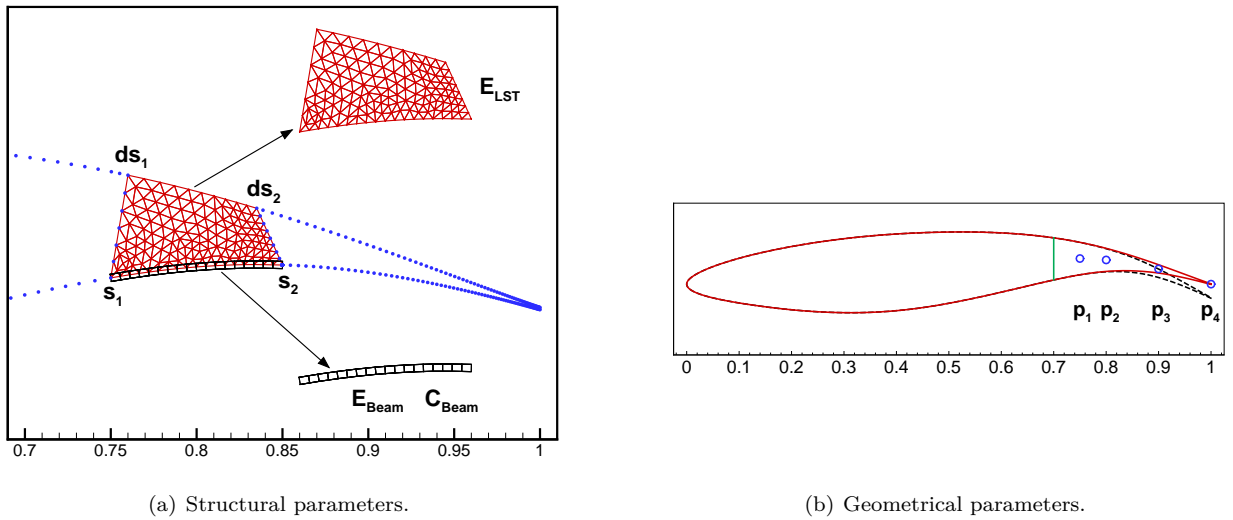


Figure 11: Illustration of hydrofoil parameters.

As mentioned before, the optimization concerns the minimization, for 4 conditions, of the hydrofoil drag coefficients $C_{D_{i=1, \dots, 4}}$. Each condition corresponds to a prescribed forward speed and hydrodynamic lift force developed by the hydrofoil. The optimization problem is written as

$$\min_{\mathbf{x} \in \Omega_{\text{adm}}} f(\mathbf{x}) = \sum_{i=1}^4 w_i C_{D_i}(\mathbf{x}), \quad (35)$$

where the $w_i > 0$ are prescribed weighting coefficients fixing the relative importance of the 4 conditions. Constrains $Q_i(\mathbf{x})$ are introduced to prevent the cavitation of the flow for 4 other conditions. The admissible and computable domain is then defined as the intersection of the set of design points \mathbf{x} satisfying the cavitation constraints, with $\Omega \setminus \Omega_{\chi}$ the set of \mathbf{x} having computable function f :

$$\Omega_{\text{adm}} = \Omega \cap \{\mathbf{x}, Q_{i=1, \dots, 4}(\mathbf{x}) \leq 0\} \setminus \Omega_{\chi}, \quad (36)$$

Indeed, parts of the search domain Ω corresponds to non-computable points, because of the lack of robustness of the simulation tool. As an example, Figure 12 shows a hydrofoil corresponding to a non-computable point of the search space, at which the flow solver is not able to determine the hydrodynamic forces. Although it

may be possible to finely tune the solvers to improve their robustness, automating the tuning process can be a difficult task. In addition, these non-computable configurations often correspond to uninteresting or even unphysical design points, such that the direct classification of these non-computable areas has been found very effective.

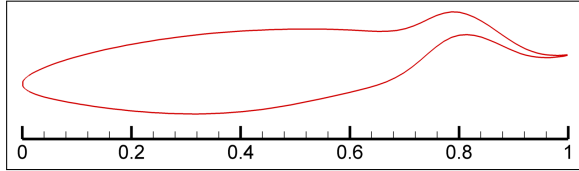


Figure 12: Configuration of the hydrofoil leading to a non computable case.

The explicit constraints Q_i are expressed as

$$Q_i(\mathbf{x}) = -C_{P_i}(\mathbf{x}) - \lambda_i, \quad \lambda_i = \frac{\bar{p}_i - p_v}{\frac{1}{2}\rho U_i^2}, \quad (37)$$

where ρ is the fluid density, p_v the saturated vapor pressure, C_{P_i} the minimal pressure coefficient, λ_i the cavitation number, and U_i and \bar{p}_i are the reference velocity and pressure for the i -th condition. The minimum of the pressure coefficient is defined by $C_{P_i}(\mathbf{x}) := \frac{p_i^-(\mathbf{x}) - \bar{p}_i}{0.5\rho U_i^2}$, where p_i^- is the lowest pressure over the hydrofoil surface. The condition $Q_i \leq 0$ expresses the constraint that the minimum of the pressure around the hydrofoil should remain higher than the vapor pressure: $p_i^-(\mathbf{x}) \geq p_v$.

Given a value \mathbf{x} of the design variable, the evaluation of the objective function f and the constraints Q_i requires the resolution of 8 nonlinear fluid-structure interaction problems. For the flow, we rely on a static vortex lattice method with viscous boundary layer equations, using the solver XFOIL [43, 44], while a nonlinear elasticity solver [45] is used for the elastic deformations of the foil [45], modeling the elastic trailing edge with 2D Linear Strain Triangles (LST) [46] and Timoshenko beam elements (see Figure 11(a)). The nonlinear equilibrium FSI solutions are computed by a Newton method with Aitken relaxation. We mention that the conditions call for the enforcement of a prescribed lift force. This is achieved in XFOIL by adjusting the Angle of Attack (AoA) of the hydrofoil.

4.2. Optimization results

In the following, we consider two EGO approaches to handle the constraints related to the cavitation. First, the proposed EGO-LS-SVM method is used to estimate directly Ω_{adm} from the classification of previous points \mathbf{x} . In other words, the LS-SVM method is employed to predict both the satisfaction of the constraints $Q_i(\mathbf{x}) \leq 0$ and the computability of future points. For the second method, the LS-SVM classification is used only to predict the computability, while the functionals Q_i of the cavitation constraints are approximated by individual GP models following the EGO method presented in [28]. We call this second approach the EGO-GPC. These two approaches are also compared to the reference CMA-ES algorithm [23] with a penalty value $f_\infty = 100$ for the non-computable points. In all computations, the two EGO methods are initialized using an LHS set of size 100. For this example, the EGO optimizations are continued until a maximum computational budget of 36 hours is exhausted. Both the resolutions of the non-linear FSI problems, the construction of the GP models and the search for the maximizer of the AEI criteria are included in the computational time. Note that on a classical workstation computer it takes approximately $T_f = 75\text{s}$ to decide the admissibility of a point and evaluate f at a given computable \mathbf{x} , solving the FSI problems corresponding to the prescribed 8 conditions.

Figure 13 shows the evolution of the successive values of f with the number of calls to f . It is seen that the EGO-LS-SVM and EGO-GPC methods require a significantly lower number of function evaluations than CMA-ES to reach the optimum neighborhood. Specifically, the CMA-ES optimization needs 1615 function evaluations, that is, twice as much, to eventually produce a better point than the EGO-LS-SVM optimum (stopped after 36h of computations). The two EGO methods have similar behavior, although the sequence of

computed values of f produced the EGO-GPC method is slightly more dispersed than for the EGO-LS-SVM method. This difference can be explained by the additional evaluations needed to construct the GP models of the constraints in the EGO-GPC method.

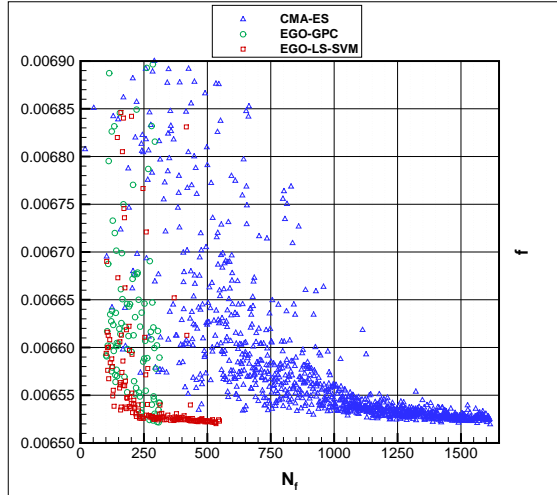


Figure 13: Sequence of computable function values $f(\mathbf{x}_n)$ as a function of the number of calls N_f for the EGO-LS-SVM, EGO-GPC and CMA-ES methods. Case of problem (35).

If the two EGO methods behave similarly in term of the convergence with the number of calls to f , they do present significant differences in terms of computational times. This can be appreciated from the plots of Figure 14. Figure 14(a) reports the cumulated computational times of the EGO-LS-SVM and EGO-GPC methods as functions of the optimization iteration. The computational times are split into the time dedicated to the FSI problems resolution, called FSI time, and the construction of the GP models and search of maximizer of the extended AEI, called EGO time. It is seen from the graph that the EGO time is dominant for the two approaches and increases with the iteration index at a polynomial rate $r \gtrsim 2$, because of the increasing size of the GP models construction. Furthermore, the EGO time of the EGO-GPC method is significantly higher than that of the EGO-LS-SVM method, as one may have expected since more GP processes must be estimated. In contrast to the EGO time, the computational times spent on the FSI problems increase roughly at a linear rate, with the EGO-LS-SVM FSI time being half of the EGO-GPC FSI time. A closer inspection of the results reveals that, when the computational budget of 36 hours is reached, the EGO-GPC method called for 312 evaluations of f , including 8 non-computable points, while EGO-LS-SVM made up to 554 calls to f , among which 294 violated the constraints and 5 were not computable. These numbers explain the lower FSI time of the EGO-LS-SVM method, compared to the EGO-GPC method. Indeed, the full classification approach of the EGO-LS-SVM method allows for significant computational savings because once the constraint on cavitation for a certain condition has been found to be non-satisfied, there is no need to solve the remaining FSI problems for the other conditions and to compute f . For the EGO-GPC method, on the contrary, one must solve systematically all the FSI problems for the whole set of 8 (constraints and objective) conditions to improve the GP models of all the Q_i and f . To further illustrate this point, we compare in Figure 14(b) the current best value of f observed, denoted $\min(f)$, as a function of the computational time, for the two EGO methods. The EGO-LS-SVM method is seen to be more efficient as it gives a better optimum than the EGO-GPC approach. Moreover, the convergence rate seems faster for the EGO-LS-SVM method: an excellent minimum within less than 0.1% of the terminal value of f is obtained in just 4 hours when for the GP method a comparable point is obtained after 17 hours. These differences are consistent with the fact that the number of EGO-LS-SVM iterations performed in a given computational time is essentially twice that of the EGO-GPC: although it is proposing a significant fraction of non-feasible points, the EGO-LS-SVM method better explores the design space and takes advantage of a cheaper evaluation cost for non-feasible points.

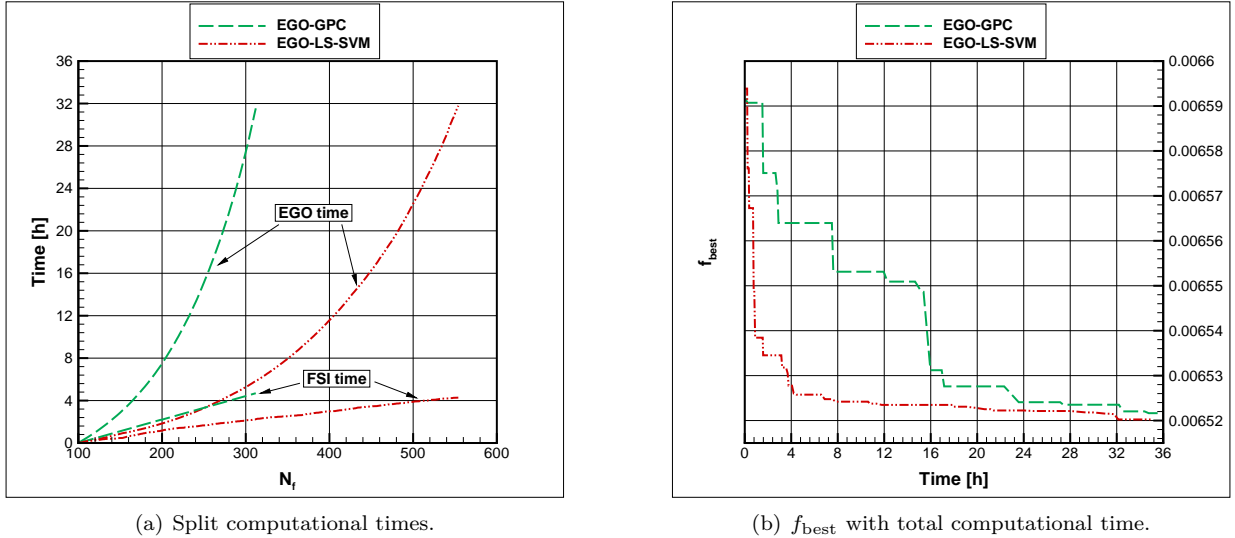


Figure 14: Left: comparison of the computational times of the EGO-LS-SVM and EGO-GPC methods, splitting the FSI solves (FSI time) and the GP models construction and AEI exploration (EGO time) contributions. Right: convergence of the computable f_{best} value with the total computational time. Case of problem (35).

Nevertheless, it should be noted that the CMA-ES algorithm can be computationally cheaper than the EGO methods, depending on the computational budget and the problem considered. For instance, Figure 15(a) shows the best admissible (and computable) value of f obtained as a function of the computational time for the two EGO methods and the CMA-ES procedure. It is seen that in the case of the hydrofoil optimization with the numerical models considered here, the CMA-ES always outperforms EGO-GPC, except for a point around 17 hours of computational time. Compared to the EGO-LS-SVM, the CMA-ES is significantly less efficient at the beginning of the optimization procedure but eventually do better as the computational time increases, with a better point found by the evolutionary approach after ≈ 26 hours of calculation. Obviously, these results are for a single realization of the optimization procedures that are random in essence (through the selection / re-sampling of the population in CMA-ES and initial LHS sample in the EGO methods). A complete comparison should consider averages over multiple realizations, but the results reported in Figure 15(a) are representative of the relative behavior of the different methods. However, these relative efficiencies are highly dependent on the problem and the numerical model involved. In the present case, the important fraction of the computational time dedicated to the GP models construction is detrimental to the relative performance of the EGO methods. To illustrate this, we can estimate the computational times that would result if the resolution of the FSI problems were much more demanding, so the EGO times become relatively less significant. This is made in Figure 15(b) where we have considered the case of an evaluation of f requiring 30 minutes of computational time, instead of 75 seconds. In this situation, the CMA-ES optimization would need roughly 525 hours to find a better point than the one obtained in just 125 hours by the EGO-LS-SVM. Even the EGO-GPC method would become more competitive than the CMA-ES strategy for T_f large enough. In conclusion, the EGO-based methods are most suited to the case of costly models.

5. Conclusion

In this work, we have proposed the use of a classification approach in EGO procedures to deal with the existence of non-computable design points and inequality constraints. The key idea of the approach is to define the admissible sub-domain of the original design space consisting of points being computable and satisfying the problem constraints. The method, named EGO-LS-SVM, relies on an LS-SVM classifier to

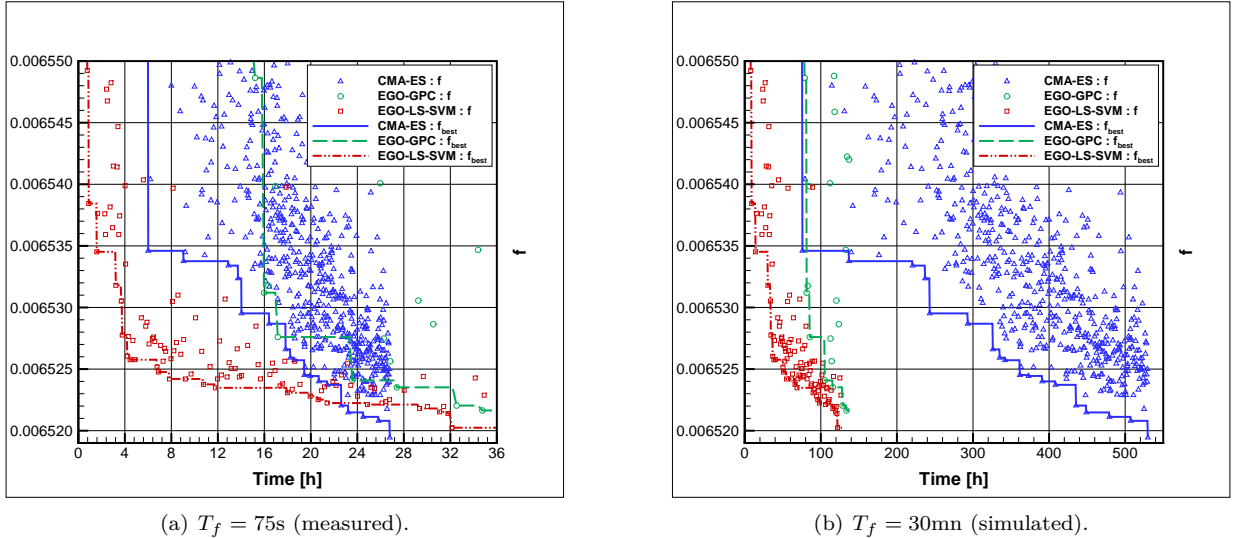


Figure 15: Evolution of the best computable objective function value f_{best} with the total computational time for the EGO-LS-SVM, EGO-GPC and CMA-ES methods and considering different evaluation times T_f . The sequence of computable function values $f(\mathbf{x}_n)$ are also reported. Case of problem (35).

predict the admissibility of points in the global search domain. The LS-SVM classifier composes a Gaussian process model with a sigmoid function, leading to an admissibility probability. It is trained on the set of previously classified points. We combine the probabilistic characterization of the admissibility with the classical Augmented Expected Improvement (AEI) statistical criterion, to propose several merit functions for the selection of the next EGO point. These merit functions induce conservative or on the contrary aggressive strategies, with respect to admissibility when exploring the design space. In fact, we recommend to actually alternate between the merit functions to yield a robust but effective sequential infilling strategy, ensuring the intensive exploration of promising areas of the search domain and the boundaries of the admissible domain.

The EGO-LS-SVM method has been tested on several analytical problems presenting different types of difficulties. These tests have validated the method and revealed its robustness with respect to the number of design variables. In fact, it is shown that compared to standard evolutionary algorithms (CMA-ES) the relative efficiency of EGO-LS-SVM is increasing with the number of parameters in the range tested. However, the EGO-LS-SVM method can be challenged in extreme situations where the admissible domain consists of a small portion of the global design set with complex geometry. For these situations, the EGO-LS-SVM may not have found admissible points, stalling the convergence.

Finally, the method was tested on the constrained optimization of a fully nonlinear numerical fluid-structure interaction system. Here we aimed at minimizing the drag of a flexible hydrofoil in several operating conditions, with constraints related to the cavitation. On this example, the EGO-LS-SVM method has been compared with the CMA-ES and EGO methods with GP models of the constraints. This example highlighted the interest of using EGO-based methods compared to the CMA-ES on these complex problems, and the advantages of treating the constraints and computable domain in a unified approach to the EGO-LS-SVM method.

Future works should focus on the current limits of the EGO-LS-SVM method. In particular, the initialization procedure must be improved in the situation where the computable domain is small compared to the whole search domain. One possible route in this direction could be to initiate the optimization with an evolutionary method, and subsequently switch to EGO-LS-SVM when sufficiently many points have been found in the admissible domain. Also, when the absolute probabilities of C^\pm are much unbalanced (small or large computable domain, compared to the search domain) the use of a Weighted-SVM approach [47] could help achieving better classification performances, in particular during the initial optimization iterations.

References

References

- [1] T. Simpson, J. Poplinski, N. P. Koch, J. Allen, Metamodels for Computer-based Engineering Design: Survey and recommendations, *Engineering with Computers* 17 (2) (2001) 129–150, ISSN 1435-5663, doi:\bibinfo{doi}{10.1007/PL00007198}, URL <http://dx.doi.org/10.1007/PL00007198>.
- [2] J. P. Kleijnen, Kriging metamodeling in simulation: A review, *European Journal of Operational Research* 192 (3) (2009) 707 – 716, ISSN 0377-2217, doi:\bibinfo{doi}{http://dx.doi.org/10.1016/j.ejor.2007.10.013}, URL <http://www.sciencedirect.com/science/article/pii/S0377221707010090>.
- [3] V. Vapnik, *The nature of statistical learning theory*, Springer-Verlag New York, Inc., New York, NY, USA, ISBN 0-387-94559-8, 1995.
- [4] A. Marrel, B. Iooss, B. Laurent, O. Roustant, Calculations of Sobol indices for the Gaussian process metamodel, *Reliability Engineering and System Safety* 94 (3) (2009) 742 – 751, ISSN 0951-8320, doi:\bibinfo{doi}{https://doi.org/10.1016/j.res.2008.07.008}, URL <http://www.sciencedirect.com/science/article/pii/S0951832008001981>.
- [5] P. Wang, Z. Lu, Z. Tang, An application of the Kriging method in global sensitivity analysis with parameter uncertainty, *Applied Mathematical Modelling* 37 (9) (2013) 6543 – 6555, ISSN 0307-904X, doi:\bibinfo{doi}{https://doi.org/10.1016/j.apm.2013.01.019}, URL <http://www.sciencedirect.com/science/article/pii/S0307904X13000413>.
- [6] H. Nickisch, C. E. Rasmussen, Approximations for binary Gaussian process classification, *Journal of Machine Learning Research* 9 (Oct) (2008) 2035–2078.
- [7] Y. Liu, Y. Shi, Q. Zhou, R. Xiu, A sequential sampling strategy to improve the global fidelity of metamodels in multi-level system design, *Structural and Multidisciplinary Optimization* 53 (6) (2016) 1295–1313, ISSN 1615-1488, doi:\bibinfo{doi}{10.1007/s00158-015-1379-9}, URL <https://doi.org/10.1007/s00158-015-1379-9>.
- [8] C. Park, R. T. Haftka, N. H. Kim, Remarks on multi-fidelity surrogates, *Structural and Multidisciplinary Optimization* 55 (3) (2017) 1029–1050, ISSN 1615-1488, doi:\bibinfo{doi}{10.1007/s00158-016-1550-y}, URL <https://doi.org/10.1007/s00158-016-1550-y>.
- [9] H. Dong, B. Song, P. Wang, Z. Dong, Surrogate-based optimization with clustering-based space exploration for expensive multimodal problems, *Structural and Multidisciplinary Optimization* ISSN 1615-1488, doi:\bibinfo{doi}{10.1007/s00158-017-1826-x}, URL <https://doi.org/10.1007/s00158-017-1826-x>.
- [10] D. R. Jones, M. Schonlau, W. J. Welch, Efficient Global Optimization of Expensive Black-Box Functions, *Journal of Global optimization* 13 (4) (1998) 455–492.
- [11] J. Liu, W.-P. Song, Z.-H. Han, Y. Zhang, Efficient aerodynamic shape optimization of transonic wings using a parallel infilling strategy and surrogate models, *Structural and Multidisciplinary Optimization* 55 (3) (2017) 925–943, ISSN 1615-1488, doi:\bibinfo{doi}{10.1007/s00158-016-1546-7}, URL <https://doi.org/10.1007/s00158-016-1546-7>.
- [12] B. Glaz, P. P. Friedmann, L. Liu, Surrogate based optimization of helicopter rotor blades for vibration reduction in forward flight, *Structural and Multidisciplinary Optimization* 35 (4) (2008) 341–363, ISSN 1615-1488, doi:\bibinfo{doi}{10.1007/s00158-007-0137-z}, URL <https://doi.org/10.1007/s00158-007-0137-z>.
- [13] B. Glaz, P. P. Friedmann, L. Liu, Helicopter Vibration Reduction throughout the Entire Flight Envelope Using Surrogate-Based Optimization, *Journal of the American Helicopter Society* 54 (1) (2009) 12007.
- [14] N. Aghajari, M. Schäfer, Efficient shape optimization for fluid–structure interaction problems, *Journal of Fluids and Structures* 57 (2015) 298 – 313.
- [15] M. Sacher, F. Hauville, R. Duvigneau, O. L. Maître, N. Aubin, M. Durand, Efficient optimization procedure in non-linear fluid-structure interaction problem: Application to mainsail trimming in upwind conditions, *Journal of Fluids and Structures* 69 (2017) 209 – 231, ISSN 0889-9746, doi:\bibinfo{doi}{http://dx.doi.org/10.1016/j.jfluidstructs.2016.12.006}, URL <http://www.sciencedirect.com/science/article/pii/S0889974616303437>.
- [16] V. Picheny, T. Wagner, D. Ginsbourger, A benchmark of kriging-based infill criteria for noisy optimization, *Structural and Multidisciplinary Optimization* 48 (3) (2013) 607–626, ISSN 1615-1488, doi:\bibinfo{doi}{10.1007/s00158-013-0919-4}, URL <https://doi.org/10.1007/s00158-013-0919-4>.
- [17] Z. Li, S. Ruan, J. Gu, X. Wang, C. Shen, Investigation on parallel algorithms in efficient global optimization based on multiple points infill criterion and domain decomposition, *Structural and Multidisciplinary Optimization* 54 (4) (2016) 747–773, ISSN 1615-1488, doi:\bibinfo{doi}{10.1007/s00158-016-1441-2}, URL <https://doi.org/10.1007/s00158-016-1441-2>.
- [18] A. K. Jain, M. N. Murty, P. J. Flynn, Data clustering: a review, *ACM computing surveys (CSUR)* 31 (3) (1999) 264–323.
- [19] Y. Zhang, C. Park, N. H. Kim, R. T. Haftka, Function Prediction at One Inaccessible Point Using Converging Lines, *Journal of Mechanical Design* 139 (5) (2017) 051402, ISSN 1050-0472, doi:\bibinfo{doi}{10.1115/1.4036130}, URL <http://dx.doi.org/10.1115/1.4036130>.
- [20] J. Suykens, J. Vandewalle, Least Squares Support Vector Machine Classifiers, *Neural Processing Letters* 9 (3) (1999) 293–300, ISSN 1573-773X, doi:\bibinfo{doi}{10.1023/A:1018628609742}, URL <http://dx.doi.org/10.1023/A:1018628609742>.
- [21] A. Basudhar, C. Dribusch, S. Lacaze, S. Missoum, Constrained efficient global optimization with support vector machines, *Structural and Multidisciplinary Optimization* 46 (2) (2012) 201–221, ISSN 1615-1488, doi:\bibinfo{doi}{10.1007/s00158-011-0745-5}, URL <http://dx.doi.org/10.1007/s00158-011-0745-5>.
- [22] T. van Gestel, J. A. Suykens, B. Baesens, S. Viaene, J. Vanthienen, G. Dedene, B. de Moor, J. Vandewalle, Benchmarking Least Squares Support Vector Machine Classifiers, *Machine Learning* 54 (1) (2004) 5–32, ISSN 1573-0565, doi:\bibinfo{doi}{10.1023/B:MACH.0000008082.80494.e0}, URL <https://doi.org/10.1023/B:MACH.0000008082.80494.e0>.

- [23] N. Hansen, The CMA Evolution Strategy: A Comparing Review, in: *Towards a new evolutionary computation*, Springer, 75–102, 2006.
- [24] C. E. Rasmussen, C. K. I. Williams, *Gaussian Processes for Machine Learning*, MIT Press, 2006.
- [25] N. Hansen, D. V. Arnold, A. Auger, Evolution strategies, in: *Springer handbook of computational intelligence*, Springer, 871–898, 2015.
- [26] N. Hansen, The CMA evolution strategy: A tutorial, arXiv preprint arXiv:1604.00772 .
- [27] D. Huang, T. T. Allen, W. I. Notz, N. Zheng, Global Optimization of Stochastic Black-Box Systems via Sequential Kriging Meta-Models, *Journal of global optimization* 34 (3) (2006) 441–466.
- [28] M. Schonlau, *Computer Experiments and Global Optimization*, Ph.D. thesis, University of Waterloo, Waterloo, Ont., Canada, Canada, aAINQ22234, 1997.
- [29] G. C. Cawley, Leave-one-out cross-validation based model selection criteria for weighted LS-SVMs, in: *The 2006 IEEE International Joint Conference on Neural Network Proceedings*, IEEE, 1661–1668, 2006.
- [30] M. Stone, Cross-Validatory Choice and Assessment of Statistical Predictions, *Journal of the Royal Statistical Society. Series B (Methodological)* 36 (2) (1974) 111–147, ISSN 00359246, doi:\bibinfo{doi}{10.2307/2984809}, URL <http://dx.doi.org/10.2307/2984809>.
- [31] V. N. Vapnik, V. Vapnik, *Statistical learning theory*, vol. 1, Wiley New York, 1998.
- [32] G. C. Cawley, N. L. Talbot, Efficient leave-one-out cross-validation of kernel fisher discriminant classifiers, *Pattern Recognition* 36 (11) (2003) 2585 – 2592, ISSN 0031-3203, doi:\bibinfo{doi}{http://dx.doi.org/10.1016/S0031-3203(03)00136-5}, URL <http://www.sciencedirect.com/science/article/pii/S0031320303001365>.
- [33] G. C. Cawley, N. L. Talbot, Fast exact leave-one-out cross-validation of sparse least-squares support vector machines, *Neural Networks* 17 (10) (2004) 1467 – 1475, ISSN 0893-6080, doi:\bibinfo{doi}{https://doi.org/10.1016/j.neunet.2004.07.002}, URL <http://www.sciencedirect.com/science/article/pii/S0893608004001431>.
- [34] G. C. Cawley, N. L. Talbot, Preventing over-fitting during model selection via Bayesian regularisation of the hyperparameters, *Journal of Machine Learning Research* 8 (Apr) (2007) 841–861.
- [35] D. M. Allen, The relationship between variable selection and data augmentation and a method for prediction, *Technometrics* 16 (1) (1974) 125–127.
- [36] B. Van Calster, J. Luts, J. A. K. Suykens, G. Condous, T. Bourne, D. Timmerman, S. Van Huffel, *Artificial Neural Networks – ICANN 2007: 17th International Conference*, Porto, Portugal, September 9-13, 2007, Proceedings, Part II, chap. Comparing Methods for Multi-class Probabilities in Medical Decision Making Using LS-SVMs and Kernel Logistic Regression, Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN 978-3-540-74695-9, 139–148, doi:\bibinfo{doi}{10.1007/978-3-540-74695-9-15}, URL <http://dx.doi.org/10.1007/978-3-540-74695-9-15>, 2007.
- [37] J. C. Platt, Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods, *Advances in large margin classifiers* 10 (3) (1999) 61–74.
- [38] H.-T. Lin, C.-J. Lin, R. C. Weng, A note on Platt’s probabilistic outputs for support vector machines, *Machine Learning* 68 (3) (2007) 267–276, ISSN 1573-0565, doi:\bibinfo{doi}{10.1007/s10994-007-5018-6}, URL <http://dx.doi.org/10.1007/s10994-007-5018-6>.
- [39] D. V. Arnold, N. Hansen, A (1+1)-CMA-ES for Constrained Optimisation, in: T. Soule, J. H. Moore (Eds.), *GECCO, ACM, ACM Press, Philadelphia, United States*, 297–304, doi:\bibinfo{doi}{10.1145/2330163.2330207}, URL <https://hal.inria.fr/hal-00696268>, 2012.
- [40] Y. Tenne, C. Goh, *Computational Intelligence in Expensive Optimization Problems, Adaptation, Learning, and Optimization*, Springer Berlin Heidelberg, ISBN 9783642107016, 2010.
- [41] J. C. Platt, *Advances in Kernel Methods*, in: B. Schölkopf, C. J. C. Burges, A. J. Smola (Eds.), *Fast Training of Support Vector Machines Using Sequential Minimal Optimization*, MIT Press, Cambridge, MA, USA, ISBN 0-262-19416-3, 185–208, URL <http://dl.acm.org/citation.cfm?id=299094.299105>, 1999.
- [42] M. Sacher, M. Durand, E. Berrini, F. Hauville, R. Duvigneau, O. Le Maître, J. A. Astolfi, Flexible hydrofoil optimization for the 35th America’s Cup with constrained EGO method, in: *International Conference on Innovation in High Performance Sailing Yachts, Innov’Sail*, 193–205, 2017.
- [43] M. Drela, XFOIL: An Analysis and Design System for Low Reynolds Number Airfoils, Springer Berlin Heidelberg, Berlin, Heidelberg, ISBN 978-3-642-84010-4, 1–12, doi:\bibinfo{doi}{10.1007/978-3-642-84010-4_1}, URL http://dx.doi.org/10.1007/978-3-642-84010-4_1, 1989.
- [44] J. Morgado, R. Vizinho, M. Silvestre, J. Páscoa, {XFOIL} vs {CFD} performance predictions for high lift low Reynolds number airfoils, *Aerospace Science and Technology* 52 (2016) 207 – 214, ISSN 1270-9638, doi:\bibinfo{doi}{http://dx.doi.org/10.1016/j.ast.2016.02.031}, URL <http://www.sciencedirect.com/science/article/pii/S1270963816300839>.
- [45] M. Durand, A. Leroyer, C. Lothodé, F. Hauville, M. Visonneau, R. Floch, L. Guillaume, FSI investigation on stability of downwind sails with an automatic dynamic trimming, *Ocean Engineering* 90 (2014) 129–139.
- [46] P. Pedersen, Some properties of linear strain triangles and optimal finite element models, *International Journal for Numerical Methods in Engineering* 7 (4) (1973) 415–429, ISSN 1097-0207, doi:\bibinfo{doi}{10.1002/nme.1620070402}, URL <http://dx.doi.org/10.1002/nme.1620070402>.
- [47] X. Yang, Q. Song, A. Cao, Weighted support vector machine for data classification, in: *Proceedings. 2005 IEEE International Joint Conference on Neural Networks*, 2005., vol. 2, ISSN 2161-4393, 859–864, 2005.